



**Universidad**  
Zaragoza



Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza

## **PROYECTO FIN DE CARRERA**

Ingeniería de Telecomunicación

Especialidad Telemática

# **SMART CITY: SISTEMA PARA ACERCAR EL PEQUEÑO COMERCIO AL CIUDADANO (MyCity4Me)**

**Jesús Martín González**

Director:

**José M<sup>a</sup> Saldaña Medina**

Ponente:

**Emiliano Bernués del Río**

Dpto. de Ingeniería Electrónica y Comunicaciones

Junio de 2014



## Resumen del proyecto

El objetivo de este proyecto es el desarrollo de un sistema, basado en una plataforma en la nube (*“cloud”*), que permite a los pequeños comercios publicar ofertas especiales de sus productos, con un periodo de vida limitado. Estas ofertas llegan a los ciudadanos a través de una aplicación instalada en su dispositivo móvil. De esta forma, se potencian las relaciones entre los ciudadanos y los pequeños comercios de una ciudad, con el fin de promover la economía, el ocio y la cultura. El sistema se compone de una aplicación web, una aplicación móvil y una infraestructura de datos en la nube.

En primer lugar, los pequeños comercios introducen las ofertas a través de una **aplicación web** que las almacena en una base de datos centralizada. Además de cargar ofertas en el sistema, el comercio puede después modificarlas o borrarlas, así como ver los resultados de las ofertas solicitadas y vendidas. Ejemplos de ofertas podrían ser: *“2x1 en entradas para el teatro entre las 19 y las 21 horas”*, *“Vino y tapa por 1€ de 13 a 14h”* o *“30% de descuento en ropa para niños”*.

La **aplicación móvil**, desarrollada para Android, permite visualizar en tiempo real las ofertas activas en la ciudad, organizadas según el tipo de negocio al que pertenezcan y en función de la posición geográfica actual del usuario. También, mediante un mapa, es capaz de mostrar los comercios que tienen ofertas validas anunciadas, de una manera más visual. Una vez seleccionando el comercio en el mapa, el usuario puede acceder a un menú con sus ofertas. La aplicación también permite ver la ruta desde su posición hasta la del comercio anunciante, así como dar una valoración a los comercios. El usuario podrá crear un perfil, para así recibir las ofertas de su interés.

La **infraestructura de datos en la nube** se ha desplegado en la plataforma FI-WARE, un servicio *“cloud”* promovido desde la Comisión Europea a través del programa FI-PPP, y en el que participan las principales empresas del sector. Se han utilizado máquinas virtuales en FI-WARE, y los componentes de software denominados *Generic Enabler*, que permiten automatizar algunas tareas.

Este sistema podrá ser explotado por todo tipo de comercios, tanto bares y restaurantes como museos, teatros, gimnasios, piscinas, tiendas de ropa, etc., suponiendo una valiosa herramienta para el turismo, que potenciará la visibilidad de los comercios, acercándolos a los ciudadanos.



# Índice

Resumen del proyecto .....	1
Índice .....	3
Agradecimientos .....	7
PARTE I MEMORIA .....	9
1    Introducción.....	11
1.1    Estructura del documento .....	11
1.2    Contexto profesional.....	12
1.3    Contexto tecnológico .....	13
1.3.1    Plataforma FI-WARE .....	14
1.3.2    Apache, PHP y MySQL .....	15
1.3.3    Android.....	15
1.3.4    HTML5 y CSS.....	17
1.3.5    XML (Extensible Markup Language) .....	17
1.3.6    JSON (JavaScript Object Notation).....	18
1.4    Motivación y objetivos .....	18
2    Trabajo realizado: <i>MyCity4Me</i> .....	21
2.1    Infraestructura de datos en la nube: FI-WARE.....	21
2.1.1    Instalación, configuración y uso combinado de los Generic Enabler.....	22
2.1.2    Análisis e implementación de la estructura de la base de datos. ....	25
2.2    Aplicación web .....	27
2.2.1    Diseño de la estructura de la aplicación web.....	28
2.2.2    Metodología de desarrollo de la aplicación web .....	31
2.2.3    Decisiones de diseño adoptadas .....	32
2.3    Aplicación Android.....	34
2.3.1    Aplicaciones relacionadas .....	34

2.3.2	Análisis de la aplicación .....	35
2.3.3	Diseño de la aplicación .....	36
2.3.4	Metodología de desarrollo de la aplicación .....	39
2.3.5	Decisiones de diseño adoptadas .....	40
2.3.6	Estructura de la aplicación .....	46
3	Conclusiones y líneas futuras .....	49
3.1	Conclusiones .....	49
3.2	Líneas futuras .....	50
3.3	Valoración personal .....	50
	REFERENCIAS .....	53
	ACRÓNIMOS .....	55
	FIGURAS .....	57
	PARTE II ANEXOS .....	61
	ANEXO A - Manual de usuario de la aplicación Android <i>MyCity4Me</i> .....	63
	A.1 ¿Qué es <i>MyCity4Me</i> ? .....	63
	A.2 Inicio de la aplicación y acceso a ella .....	63
	A.3 Consulta y solicitud de las ofertas activas en la ciudad .....	66
	ANEXO B - Manual de usuario de la aplicación web .....	71
	B.1 Modo administrador .....	71
	B.2 Modo comercio .....	75
	ANEXO C - Fi-Lab, entorno de desarrollo de FI-WARE .....	83
	C.1 ¿Qué es FI-Lab? .....	83
	C.2 Su uso en el presente proyecto: creación de instancias o máquinas virtuales. ....	84
	ANEXO D - Puesta en marcha del sistema <i>Orion-Cygnus-Cosmos</i> .....	89
	ANEXO E - Implementación del servicio web PHP .....	95
	ANEXO F - Elementos típicos de una aplicación Android .....	97
	ANEXO G - Diseño y navegación de la aplicación Android .....	99

ANEXO H - Activación de los mapas de Google en MyCity4Me.....	101
ANEXO I - Metodología de desarrollo de la aplicación Android .....	103
I.1 Diseño gráfico de la ventana .....	103
I.2 Análisis, obtención y presentación de la información.....	104
I.3 Dotación de interacción.....	107
I.4 Pruebas de integración en la aplicación .....	108
ANEXO J - Metodología de desarrollo de la aplicación web .....	111
J.1 Diseño gráfico de la ventana .....	111
J.2 Análisis, obtención y presentación de la información .....	112
J.3 Dotación de interacción .....	113
J.4 Pruebas.....	114
ANEXO K - Herramientas utilizadas. ....	115





## **Agradecimientos**

En primer lugar quisiera agradecer todo el apoyo y la ayuda de las personas más cercanas, es decir, a la familia, que siempre está ahí, a las buenas y a las malas, que ha sido un apoyo constante, y a mis amigos, que cuando los necesitas, ahí están, algo que siempre es de agradecer. En especial me gustaría dar las gracias a Carlos Fernández, por prestarme sus conocimientos acerca de la programación web y a Diego melero, por hacer lo que mejor se le da, el diseño grafico del proyecto te lo debo a ti, crack!

En segundo lugar me gustaría dar las gracias a todos los miembros del Servicio de Ciudad Inteligente de Zaragoza, por brindarme la magnífica oportunidad de hacer este proyecto y por lo bien que me han tratado, haciéndome sentir en todo momento como uno más de este gran equipo. Especialmente me gustaría agradecer a Daniel Sarasa, mi jefe, y a Ana Jiménez, mi compañera, todo el conocimiento y apoyo que me han dado, sois geniales, gracias.

Por último, y no por ello menos importante, me gustaría dar las gracias a todos los profesores que han aportado algo a mi formación, como persona y como ingeniero. En especial, a mi tutor y guía durante toda la carrera, Ignacio Martínez, gracias a quien estoy ahora aquí, y a mis dos tutores de proyecto, Emiliano Bernués y José María Saldaña, dos grandes personas que me han ayudado, apoyado y guiado a lo largo de este proyecto, sin importar el momento y siempre con buen humor, gracias a vosotros he conseguido llevar a cabo este magnífico proyecto.

Sinceramente, gracias a todos.



# **PARTE I**

## **MEMORIA**



# 1 Introducción

El presente documento tiene como objetivo recoger toda la información acerca de la realización de este Proyecto Fin de Carrera (PFC), titulado “*Smart City: Sistema para acercar el pequeño comercio al ciudadano (MyCity4Me)*”.

En este primer capítulo se define la estructura del documento y el contexto, tanto profesional como tecnológico, en el que se ha desarrollado el proyecto, así como la motivación por la que se ha decidido llevarlo a cabo, y los objetivos iniciales planteados.

## 1.1 Estructura del documento

El presente documento está dividido en dos partes. La primera parte está dedicada a la Memoria del PFC, y pretende aportar una visión global del proyecto sin entrar en demasiados detalles de bajo nivel, y presentar las conclusiones a las que se ha llegado.

La segunda parte está comprendida por anexos que detallan con mayor precisión diferentes aspectos del proyecto realizado, del proceso que se ha llevado a cabo y de los resultados obtenidos.

- **Parte I. Memoria**

- **Capítulo 1. Introducción.** Presenta el contexto del presente trabajo.
- **Capítulo 2. Trabajo Realizado: *MyCity4Me*.** En este capítulo se ha detallado el análisis, la implementación y el desarrollo del sistema, concretamente de cada una de las partes que lo forman.
  - Infraestructura de datos en la nube: FI-WARE.
  - Aplicación web.
  - Aplicación Android.
- **Capítulo 3. Conclusiones.** Capítulo dedicado a las conclusiones obtenidas con la realización del PFC, tanto de carácter profesional como personal. Además se evaluará el cumplimiento de los objetivos y se expondrán las posibles líneas futuras de trabajo que pueden seguirse gracias a los resultados obtenidos.
- **Referencias**

- **Acrónimos**
- **Índice de Figuras**
- **Parte II. Anexos**
  - **Anexo A - Manual de usuario de la aplicación Android, *MyCity4Me*.**
  - **Anexo B - Manual de usuario de la aplicación web.**
  - **Anexo C - Fi-Lab, entorno de desarrollo de FI-WARE.**
  - **Anexo D - Puesta en marcha del sistema Orion-Cygnus-Cosmos.**
  - **Anexo E - Implementación del servicio web PHP.**
  - **Anexo F - Elementos típicos de una aplicación Android.**
  - **Anexo G - Diseño y navegación de la aplicación Android.**
  - **Anexo H - Activación de los mapas de Google en *MyCity4Me***
  - **Anexo I - Metodología de desarrollo de la aplicación Android.**
  - **Anexo J - Metodología de desarrollo de la aplicación web.**
  - **Anexo K - Herramientas utilizadas.**

## **1.2 Contexto profesional**

Este PFC se ha llevado a cabo en colaboración con el *Servicio de Ciudad Inteligente* del Ayuntamiento de Zaragoza, departamento especializado en la elaboración de proyectos de *Smart City*. Este término se utiliza para referirse a las ciudades inteligentes, que utilizan las TIC (Tecnologías de la Información y las Comunicaciones) para dar un mejor servicio al ciudadano y gestionar sus servicios de una manera más eficiente.

Toda ciudad inteligente debe apoyarse en una plataforma tecnológica adecuada. En los últimos años, la Comisión Europea, dentro del programa FI-PPP (*Future Internet Public-Private-Partnership*) [1], ha creado la plataforma FI-WARE [2], con el objetivo de servir como soporte para las ciudades inteligentes europeas. Se trata de una plataforma de muy reciente desarrollo, que permite desplegar servicios en la nube (generalmente conocida como “*cloud*”) [3] y flexibilizar así el soporte a los clientes.

En este contexto, el Ayuntamiento busca la forma de ayudar al pequeño comercio de la ciudad mediante el uso de las TIC. De aquí surge una idea de crear un sistema en el que los comercios introduzcan sus ofertas en forma de anuncios, que lleguen fácilmente a los ciudadanos por medio de una aplicación para dispositivos móviles. Dada la

popularidad de los *smartphone*, se ha considerado que una aplicación móvil puede suponer un medio muy adecuado para llegar a los ciudadanos.

De la unión de estos dos hechos surge este PFC, es decir, de la oportunidad de poner a prueba la tecnología FI-WARE para desarrollar un sistema útil para el pequeño comercio. De esta manera se cumplen al mismo tiempo los dos objetivos propuestos por el *Servicio de Ciudad Inteligente*.

### 1.3 Contexto tecnológico

Antes de entrar en detalle en el desarrollo de este sistema, es necesario introducir brevemente las tecnologías que han sido utilizadas. También se van a comentar, cuando sea necesario, las razones por las que se ha optado por una tecnología determinada.

El sistema se basa en una **aplicación móvil** instalada en los dispositivos de los clientes. Pero además debe incluir una **infraestructura de datos en la nube**, en la que se pueda almacenar la información sobre las ofertas de los comercios. Para ello se han usado tres máquinas virtuales en la *cloud* de FI-WARE. Una **aplicación web** es la encargada de recoger las ofertas de los pequeños comercios para introducirlas en la base de datos en la nube. En la Figura 1.1 se pueden observar los tres elementos que componen el sistema y las tecnologías utilizadas en cada uno, que se describen brevemente a continuación.

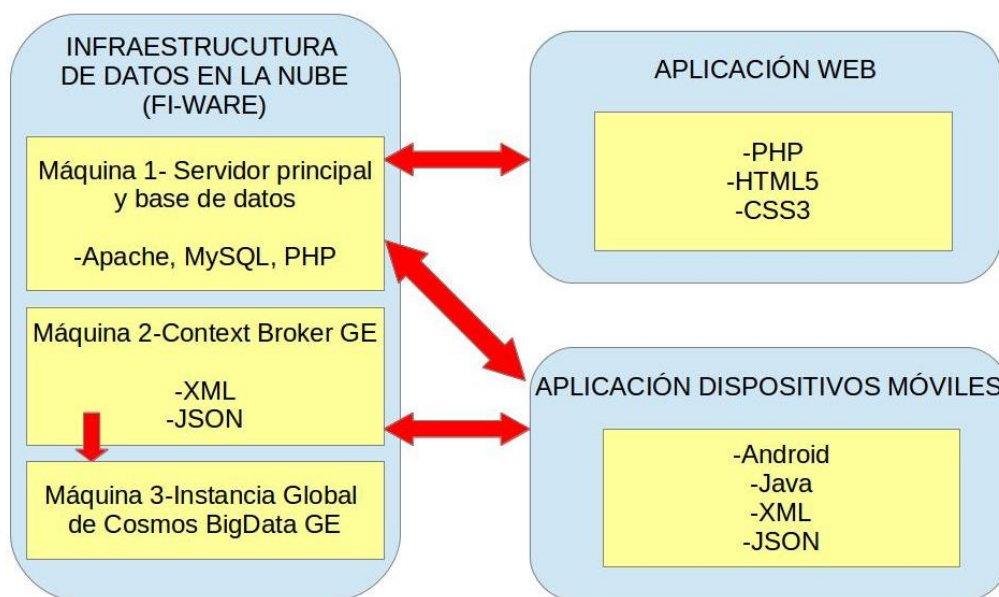


Figura 1.1. Estructura básica del sistema y las tecnologías utilizadas.

### 1.3.1 Plataforma FI-WARE

El Proyecto FI-WARE está siendo desarrollado por un consorcio formado por las mayores empresas europeas de las tecnologías de la información y las comunicaciones, con el objetivo de establecer una plataforma para la Internet del Futuro (*Future Internet*, FI). FI-WARE se puede definir [4] como “una infraestructura innovadora basada en la nube, abierta para la creación y difusión de aplicaciones y servicios futuros de Internet, de forma rentable y a una escala nunca antes vista”.

FI-WARE ofrece un servicio web en el que desplegar máquinas virtuales sobre las que trabajar remotamente. Es un servicio comparable al ofrecido comercialmente por algunas empresas, por ejemplo el *Elastic Compute Cloud* (EC2)<sup>1</sup> de Amazon, en el que los clientes pueden reservar recursos en la nube, con un coste derivado de la capacidad de las máquinas virtuales y del tiempo que se han usado. FI-WARE proporciona además una mejora importante, pues pone a disposición del usuario gran variedad de recursos y herramientas (denominadas *Generic Enabler*) que resultan útiles en el desarrollo, por ejemplo, de aplicaciones o sistemas de tratamiento de datos. El hecho de que esta infraestructura sea innovadora, y que tenga el apoyo de la Unión Europea, hace que su uso y aprendizaje resulten atractivos, dada la proyección de futuro que puede tener.

El entorno de trabajo de FI-WARE se conoce como FI-Lab [5]. Se trata de una instancia de FI-WARE a la que se puede acceder para realizar pruebas y experimentos que requieran recursos de la nube. Por su carácter experimental, hasta el momento se puede utilizar de forma gratuita. Además, tiene la finalidad de ser un medio de comunicación entre desarrolladores de aplicaciones y posibles empresas o inversores interesados en realizar desarrollos sobre FI-WARE.

Esta plataforma cuenta con un conjunto de componentes, llamados *Generic Enabler* [6], que se definen como “*bloques funcionales formados por un conjunto de elementos que juntos soportan un conjunto determinado de funciones y ofrecen un conjunto determinado de APIs e interfaces interoperables que cumplen con las especificaciones abiertas publicadas para ese GE*”. Estos componentes permiten a los desarrolladores:

- Crear, publicar administrar y consumir servicios de *Future Internet*.
- Desplegar los servicios de *Future Internet* en la nube.

---

<sup>1</sup> La propia Amazon lo define como “un servicio web que proporciona capacidad informática con tamaño modificable en la nube. Está diseñado para facilitar a los desarrolladores recursos informáticos escalables basados en web” (ver <https://aws.amazon.com/es/ec2/>).



- Acceder, procesar y analizar grandes cantidades de datos.
- Aprovechar la ubicuidad de dispositivos heterogéneos y de uso restringido en el Internet de las Cosas.
- Acceder a redes y dispositivos a través de interfaces de servicio.

En este proyecto se han usado dos *Generic Enabler*, cuya función es almacenar las coordenadas y la fecha exacta en la que los usuarios se conectan a la aplicación del móvil. Los *Generic Enabler* encargados de esta labor son: *Orion Context Broker* [7] y *Cosmos - BigData Analysis* [8], cuyo funcionamiento detallado se explicará más adelante, en el apartado 2.1.1.

### 1.3.2 Apache, PHP y MySQL

Apache es un servidor web HTTP de código abierto para plataformas Unix, Windows, Apple y otros, que implementa el protocolo HTTP/1.12. Es utilizado principalmente para enviar páginas web estáticas o dinámicas en la *World Wide Web*. Es el componente de servidor web en la popular plataforma de aplicaciones LAMP (Linux-Apache-MySQL-PHP, Perl, Python) o XAMPP (Windows-Apache-MySQL-PHP, Perl, Python).

En este proyecto se ha utilizado el paquete formado por Apache, como servidor web, MySQL como base de datos y PHP como lenguaje de programación, que se instala en las máquinas virtuales Linux de FI-WARE como núcleo del sistema. Además, para facilitar el trabajo con la base de datos, se instala una herramienta, denominada *PHPMysqlAdmin*<sup>2</sup>, que añade una interfaz visual que permite realizar todo tipo de tareas sobre MySQL (crear, modificar y borrar bases de datos, tablas y columnas, etc.) sin tener que realizarlas a nivel de comandos.

### 1.3.3 Android

Android es un sistema operativo para dispositivos móviles. Está desarrollado por la *Open Handset Alliance*, una alianza comercial de 78 compañías importantes, como Samsung, HTC, Dell, Intel, etc., lideradas por Google, para desarrollar estándares abiertos para dispositivos móviles.

---

<sup>2</sup> PHP MyAdmin, Bringing MySQL to the web, [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php) [accedido junio 2014]

El sistema operativo Android se compone de aplicaciones que se ejecutan en un *framework* Java de aplicaciones orientadas a objetos, sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik, definida en [9], con compilación en tiempo de ejecución.

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. Es el lenguaje más popular en la comunidad de desarrolladores, y ésta es una de las razones por las que el desarrollo de aplicaciones Android se ha puesto de moda en los últimos años.

En Android, las aplicaciones desarrolladas para una versión concreta del sistema operativo son compatibles con dicha versión y las posteriores. Es decir, a menor versión de la aplicación desarrollada, mayor será el número de dispositivos móviles con los que será compatible.

El objetivo es llegar al máximo número de dispositivos móviles, por lo que la versión ideal para el desarrollo del PFC sería en principio la 2.3 (*Gingerbread*). Sin embargo, los requisitos de algunos métodos utilizados en la aplicación nos han obligado a establecer como versión mínima la 4.0 (*Ice Cream Sandwich*). En la Figura 1.2 se puede observar el porcentaje de uso que tenía cada una de las versiones existentes en el momento del inicio del proyecto. Se puede observar que el 80% de los móviles Android actuales son capaces de ejecutar nuestra aplicación.

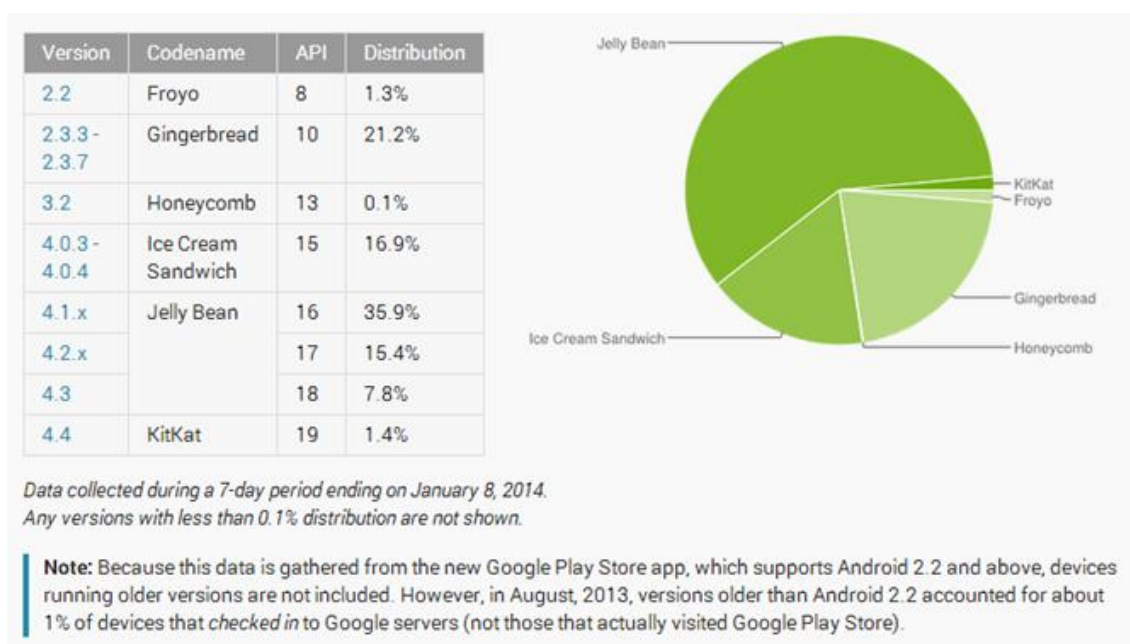


Figura 1.2. Porcentaje de uso de las versiones Android. Enero 2014 [10]

Se ha decidido usar Android ya que tiene una gran aceptación entre los usuarios de *smartphone*. Tal y como se puede apreciar en la Figura 1.3, Android (30.23%), con sus distintas versiones, está por encima de *iPhone* (22.36%).

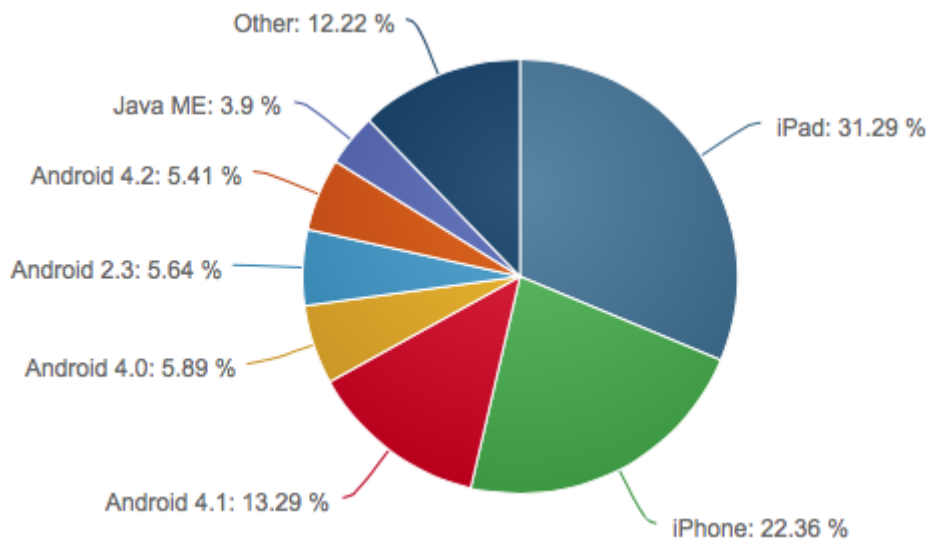


Figura 1.3. Distribución del uso de sistemas operativos para dispositivos móviles [11]

### 1.3.4 HTML5 y CSS

HTML5 (*HyperText Markup Language*, versión 5) es la quinta revisión del lenguaje básico de la *World Wide Web*. HTML se define como el estándar de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición del contenido de una página web.

CSS (*Cascading Style Sheets*) es el lenguaje de hojas de estilo utilizado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas, esto incluye varios lenguajes basados en XML como son XHTML o SVG.

Estos dos lenguajes, junto con PHP, son los encargados de definir la aplicación web desarrollada en este sistema.

### 1.3.5 XML (Extensible Markup Language)

XML es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C). Concretamente, en el presente proyecto tiene el papel de definir el diseño gráfico de las distintas pantallas de las que consta la aplicación móvil. También

tiene un papel muy importante en el archivo de configuración, *AndroidManifest.xml*, presente en todo proyecto Android.

Además es un estándar clásico para el intercambio de datos con un servicio web, aunque en nuestro caso se ha optado por otro estándar que está en pleno auge, JSON (*JavaScript Object Notation*). XML también va a ser utilizado para enviar información a uno de los *Generic Enabler* de FI-WARE.

### **1.3.6 JSON (JavaScript Object Notation)**

XML ha sido el estándar para el intercambio de datos entre aplicaciones, pero desde hace un tiempo, su uso está disminuyendo, debido al éxito de otro estándar de mensajes más ligero y fácil de interpretar como es JSON tal y como se explica en [12].

El papel de JSON en este proyecto se centra en el intercambio de información entre la aplicación Android y el servidor web que dispone de acceso a la base de datos principal.

Los motivos para escoger JSON son:

- Posee un formato ligero, compacto y eficaz.
- Es muy popular y está muy extendido en el desarrollo de aplicaciones Android, lo que conlleva que exista una gran cantidad de información en la red acerca de su utilización.
- Es fácil de usar y tiene soporte en el SDK de Android.

## **1.4 Motivación y objetivos**

En los últimos años los pequeños comercios han perdido cuota de mercado [13] y, de hecho, muchos han tenido que cerrar. Uno de los problemas que han llevado a esta situación es la falta de innovación. Mientras la ciudad y sus ciudadanos evolucionan y apuestan por las tecnologías, muchos comercios no lo han hecho y se han quedado atrás. Además, en los últimos años se ha incrementado el uso de los dispositivos móviles, llegando la penetración de los *smartphone* a un 55% en Europa [14]. De estos dos hechos es de donde surge la idea de este proyecto: facilitar una plataforma innovadora para publicitar las ofertas de los comercios, mediante un sistema fácil de usar y que va a llegar a los dispositivos móviles de los ciudadanos, es decir, de sus clientes. Simultáneamente, los ciudadanos se van a ver beneficiados por el sistema, pues podrán

consultar de modo sencillo las ofertas que más les convengan, según su posición geográfica o sus intereses.

Hasta ahora la mayoría de estos pequeños comercios tenían, como único medio de promocionar sus productos, los carteles en las puertas de sus establecimientos, de manera que el cliente debía pasar cerca del establecimiento para conocer sus ofertas. Con este nuevo sistema, el cliente conocerá esas ofertas tanto si está lejos, como si está en la calle paralela a la del establecimiento, ampliando así el campo de posibles clientes en una gran proporción gracias a la tecnología y de una manera muy económica.

Por otra parte, el desarrollo de este proyecto ha permitido conocer más de cerca la plataforma FI-WARE, que es utilizada como un elemento esencial, así como adquirir conocimientos acerca del desarrollo en Android, de aplicaciones web y la configuración de servidores y bases de datos.

Además de lo expuesto, existen otros motivos que han llevado al desarrollo de este proyecto:

- Ser capaz de aportar un servicio que va ser utilizado en la ciudad.
- Se trata de un proyecto innovador, pues no hay ningún sistema de estas características implantado en la ciudad y, además, las plataformas *cloud* como FI-WARE son todavía poco utilizadas.
- Como manifestación del interés de la Comisión Europea en la iniciativa FI-PPP, se han organizado concursos para estudiantes y emprendedores, con el objetivo de desarrollar aplicaciones novedosas que hicieran uso de la plataforma FI-WARE. La realización de este PFC nos ha permitido participar en uno de estos concursos, siendo seleccionada nuestra aplicación como una de las 20 finalistas del reto *Smart Business Challenge* de FI-WARE [15].

Una vez explicados los motivos, pasamos a definir cuál es el objetivo principal del proyecto, así como el resto de objetivos planteados y alcanzados durante su desarrollo.

El *objetivo principal* ha consistido en crear un sistema innovador para el pequeño comercio, en el que los comerciantes puedan introducir sus datos y ofertas diarias, y éstas lleguen a los ciudadanos mediante el uso de una aplicación móvil, de manera que puedan aprovechar esa información y disfrutar de las ofertas de los comercios que les rodean.

El ciudadano será así capaz de visualizar todas las ofertas de los comercios de su ciudad, o bien tan sólo las de su alrededor.

Para llegar al objetivo final, se han planteado durante el desarrollo del proyecto una serie de *objetivos intermedios*, que han sido los siguientes:

1. Analizar y definir cuáles son los elementos que forman el sistema de ofertas orientado al pequeño comercio.
2. Analizar el modo de desarrollar dicho sistema utilizando la plataforma FI-WARE.
3. Desplegar la infraestructura de datos en la nube de FI-WARE.
4. Desarrollar una aplicación web para que los comercios puedan hacer uso del sistema de una manera sencilla.
5. Realizar un estudio previo de la aplicación Android a desarrollar, buscando los requisitos fundamentales.
6. Desarrollar la aplicación para dispositivos móviles Android con el fin de que la información sobre ofertas llegue a los usuarios finales, los ciudadanos, basándose en su posición geográfica.
7. Integrar varios *Generic Enabler* de FI-WARE en el sistema.

## 2 Trabajo realizado: *MyCity4Me*

El sistema implementado en este proyecto está formado por

- a) una **infraestructura de datos en la nube**, donde se encuentran los servidores virtuales y la base de datos;
- b) una **aplicación web** que permite a los comercios publicar ofertas en el sistema;
- c) una **aplicación Android** para móviles que sirve para visualizar y sacar partido a las ofertas publicadas en el sistema.

A continuación se describen los tres componentes en los que se divide el sistema. La estructura se representa en la Figura 2.1, a la que haremos referencia en las siguientes secciones.

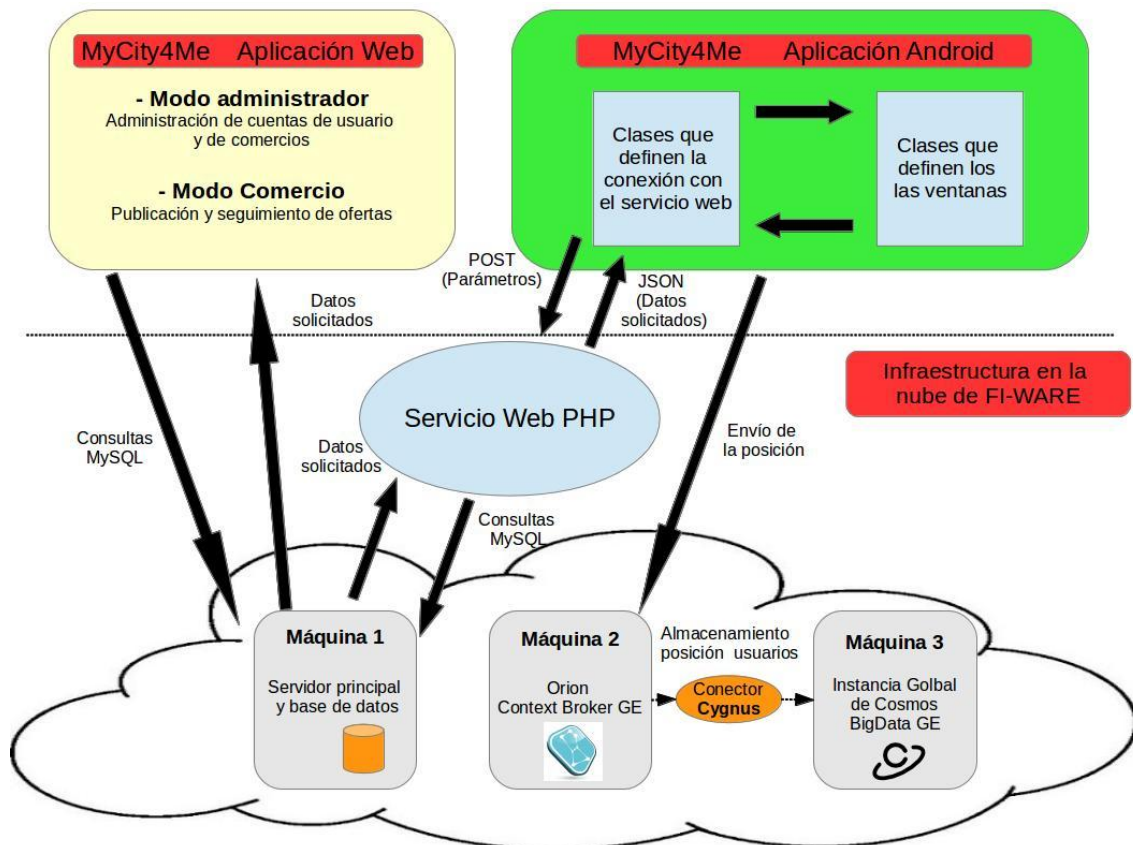


Figura 2.1. Estructura general del sistema

### 2.1 Infraestructura de datos en la nube: FI-WARE

En este apartado se va a explicar cómo se ha utilizado la plataforma FI-WARE para implementar la infraestructura de datos en la nube, describiendo cómo se ha hecho uso

de su tecnología y cómo hemos integrado nuestro sistema en ella. Al ser FI-WARE una tecnología novedosa, que continúa en proceso de desarrollo, ha sido necesario ir adaptándose a las continuas actualizaciones de software que se han ido llevando a cabo a lo largo del desarrollo del proyecto.

FI-WARE dispone de un entorno de creación denominado FI-Lab, en el que se puede configurar una infraestructura de máquinas virtuales con la que implementar la infraestructura de *cloud* necesaria para dar soporte a las aplicaciones. Para ello, se debe hacer uso de las API proporcionadas por los *Generic Enabler* de FI-WARE, desplegados o bien como un servicio global, es decir, un servicio compartido por más de un usuario, o bien como un servicio propio, creado por un único usuario y al que sólo él tiene acceso.

Para poder hacer uso de todos estos recursos en la nube es necesario desplegar servidores virtuales. El despliegue de dichos servidores virtuales se explica en el Anexo C, “FI-LAB, entorno de desarrollo de FI-WARE”.

El servicio web PHP que aparece en la Figura 2.1 se ha creado para que Android pueda tener acceso a la base de datos y su descripción detallada se encuentra en el Anexo E “Implementación del servicio web PHP”.

A continuación se explica qué *Generic Enabler* se han utilizado, así como la configuración necesaria para integrarlos en el sistema. También se describe la estructura de la base de datos creada para nuestro sistema.

### **2.1.1 Instalación, configuración y uso combinado de los Generic Enabler.**

Como ya se ha comentado antes, FI-WARE, además de ofrecer un servicio de alquiler de servidores virtuales (gratuito hasta la redacción de este documento), pone a disposición de sus usuarios una serie de herramientas denominadas *Generic Enabler* (*GE*). En este sistema se han empleado dos GE, con el objetivo de almacenar las coordenadas y la fecha en el momento en que un usuario accede a la aplicación móvil. Los GE utilizados son:

- *Orion Context Broker* GE [7]. Se encarga de gestionar información de contexto, definida en [16], es decir, información normalmente generada por sensores, y que puede ser necesaria para conocer las características de un determinado entorno. Como se ve en la Figura 2.2, algunos ejemplos de información de contexto podrían ser: el nivel de



ruido de una habitación, el nivel de luz, la capacidad de una red, la posición de un objeto, etc.

*Orion* interactúa con aplicaciones productoras de datos de contexto, que proporcionan información de sensores, y con aplicaciones consumidoras de datos de contexto, que procesan esta información para, por ejemplo, mostrarla en una gráfica.

El GE también gestiona la disponibilidad de dicha información. *Orion* permite registrar elementos de contexto y administrarlos mediante actualizaciones y consultas. Además permite suscribirse a la información de contexto de manera que se recibe una notificación cuando ocurre un determinado evento, como el transcurso de un periodo de tiempo o un cambio en alguno de los elementos.

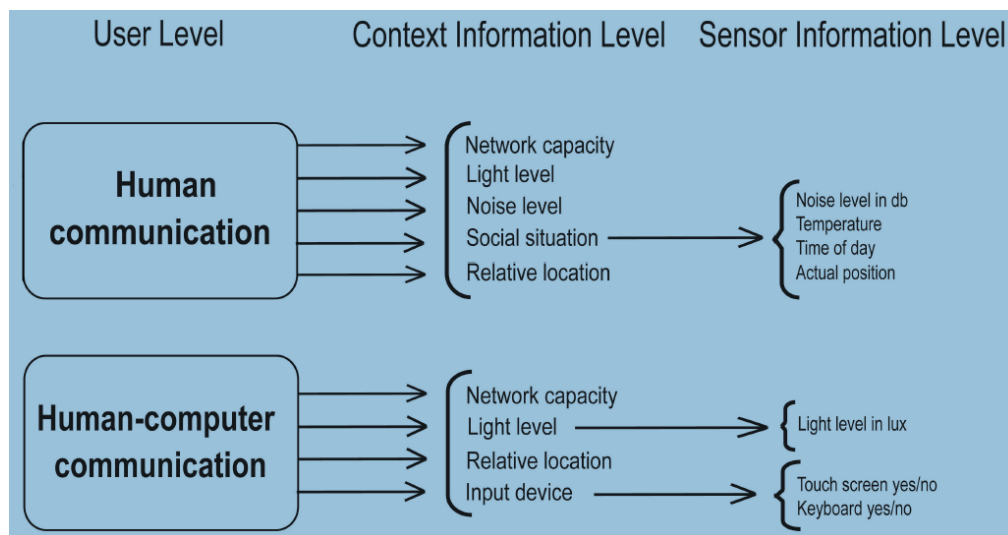


Figura 2.2. Ejemplo de información de contexto.

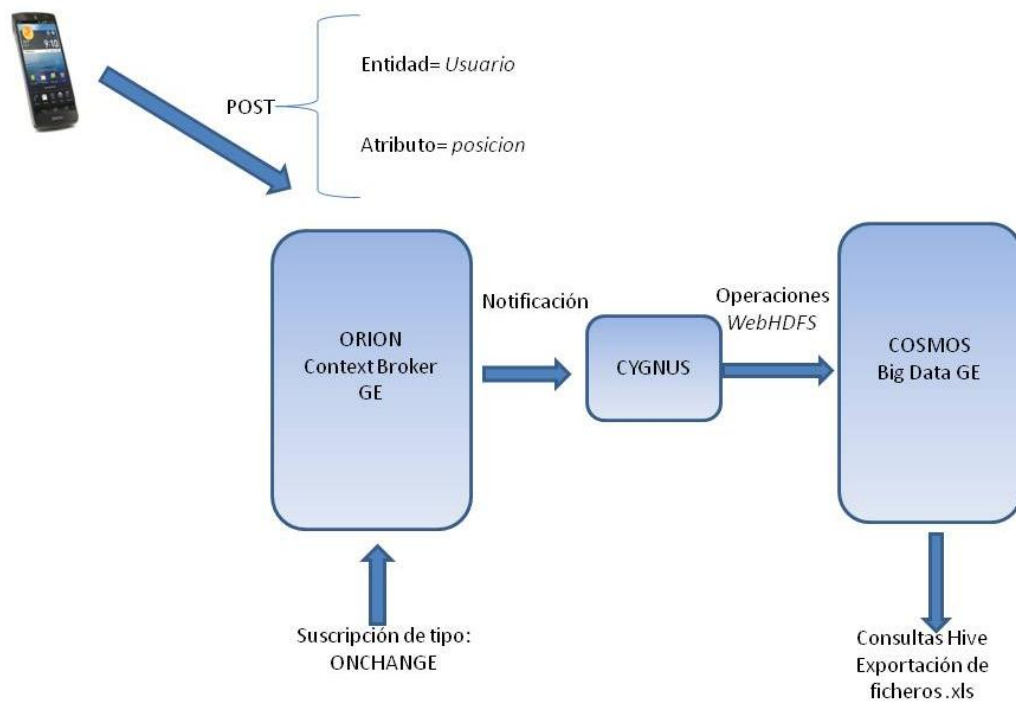
- *Cosmos - Big Data Analysis GE* [8]. Se trata de una plataforma, basada en el ecosistema *Hadoop*<sup>3</sup>, que ha sido diseñada para procesar grandes cantidades de datos (*Big Data*), en modo *batch* (por lotes), distribuyendo tareas en clústeres de ordenadores de un tamaño mediano, siguiendo el paradigma de *MapReduce* [17]. Esta plataforma proporciona el marco de trabajo necesario para que los desarrolladores se puedan centrar en resolver problemas de análisis, sin tener que preocuparse de la distribución de los datos y su sincronización. También hay un módulo encargado de recibir datos de contexto procedentes de *Orion* y de almacenarlos en *HDFS (Hadoop Distributed File System)* [18].

<sup>3</sup> <http://hadoop.apache.org/>

Una vez descritos los dos GE, se puede explicar el uso que se les da en este sistema. En nuestro caso, *Orion* se utiliza para escuchar actualizaciones de la posición geográfica de los usuarios que se conectan a la aplicación móvil y enviarlas a *Cosmos*, por medio de un conector llamado *Cygnus* [19], que está basado en *Flume* [20], almacenándolas así en su clúster *HDFS*, tal y como se representa en la Figura 2.3.

El uso de *Cosmos* es necesario debido a que *Orion* únicamente almacena la última actualización recibida de los atributos de cada entidad, por lo que si se necesita tener un historial con el valor de los atributos a lo largo del tiempo, es necesario implementar este sistema de almacenamiento.

Para poder utilizar los GE es necesario en primer lugar instalar instancias de ambos, y a continuación programar las funciones que se quiere que lleven a cabo. Este proceso de instalación y configuración se describe detalladamente en el Anexo D “Puesta en marcha del sistema *Orion-Cygnus-Cosmos*”.



*Figura 2.3. Proceso de almacenamiento de coordenadas combinando el uso de Orion y Cosmos, a través del conector Cygnus.*

Tal y como se explica en dicho anexo, se crea una entidad llamada *Usuario*, con un atributo, denominado *posición* y se hace una suscripción de esta entidad al conector *Cygnus*, el cual escucha las notificaciones ante cambios en las coordenadas recibidas y en caso de recibir una de estas notificaciones, las almacena en *Cosmos*. Posteriormente,

desde este GE se pueden realizar consultas SQL para hacer uso de los datos almacenados.

### **2.1.2 Análisis e implementación de la estructura de la base de datos.**

Un correcto diseño de la estructura de la base de datos exige analizar en primer lugar cuáles son los datos necesarios para el funcionamiento óptimo del sistema. Para realizar el diseño se han utilizado las reglas básicas de normalización de bases de datos, definidas en [21].

Como ya sabemos, el sistema integra una aplicación web, con dos modos de acceso: como administrador del sistema, únicamente accesible por esa persona, o como comercio usuario del sistema, para introducir ofertas.

Para que los comercios puedan hacer uso de la aplicación web, es necesario almacenar: *a)* los datos de acceso al sistema (nombre de usuario y contraseña); *b)* las coordenadas exactas de su localización; *c)* un perfil ligero con información sobre el comercio, incluyendo nombre, tipo de comercio, dirección y teléfono.

Es necesario también almacenar los datos de cada oferta, que son: descripción de la oferta, fecha de inicio, fecha límite, género y edad a la que va dirigida y si va dirigida a usuarios de *Tarjeta Ciudadana*<sup>4</sup> o no, tarjeta cuyo servicio se describe en [22].

Es necesario también almacenar los datos de los usuarios de la aplicación Android, introducidos al darse de alta en el sistema, así como datos generados durante el uso de la aplicación. Estos datos son: nombre de usuario, contraseña, género, fecha de nacimiento y si es usuario de *Tarjeta Ciudadana* o no. Para verificar que un usuario está en posesión de una *Tarjeta Ciudadana* activa, se utilizaría un servicio web del Ayuntamiento al que se envía el identificador de la tarjeta y éste devuelve si la tarjeta esta activa o no. Sin embargo, este servicio aún no se encuentra disponible en los servidores del Ayuntamiento, por lo que hemos tenido que simularlo, permitiendo al usuario seleccionar si tiene *Tarjeta Ciudadana* o no.

En aquellos casos en los que el usuario decida no crear una cuenta y prefiera acceder como un invitado, se almacena automáticamente un nombre de usuario del tipo “*inv0052*” y un perfil estándar.

---

<sup>4</sup> Es una tarjeta inteligente multiservicio que funciona por proximidad y permite unificar múltiples tarjetas ya distribuidas. Es personal e intransferible y permite hacer uso de servicios y micropagos.

Los datos generados por el uso de la aplicación Android son las solicitudes de ofertas, y la valoración de comercios. Una solicitud es como una reserva, pero sin cargo alguno, simplemente simboliza que un usuario le ha gustado la oferta e intentará ir a consumirla. Esto permitirá al sistema tener un histórico de los productos que interesan a cada usuario.

Además de estos datos, se almacenan una serie de identificadores, únicos e irrepetibles, denominados *claves primarias*<sup>5</sup>. Por ejemplo, a cada comercio se le asigna un identificador único que luego servirá para identificar los datos distribuidos en las distintas tablas correspondientes a ese comercio (columna *id\_bus*, Figura 2.4).

De esta manera, la base de datos queda estructurada de la siguiente forma:

- **Tabla *admin*:** Esta tabla únicamente tiene la función de almacenar el usuario y la contraseña de acceso a la web del administrador.
- **Tabla *business*:** En esta tabla se almacenan todos los datos correspondientes a los comercios integrados en el sistema, junto a la clave primaria *id\_bus*, para poder comunicarse con datos de otras tablas que tengan relación con cada comercio. También se define la columna *user* como clave primaria, ya que no puede haber dos comercios con el mismo nombre.
- **Tabla *user*:** Contiene los datos de usuario de la aplicación Android. Incluye una clave primaria denominada *id\_user*.
- **Tabla *guests*:** Contiene los datos de usuarios de tipo invitado. Incluye como clave primaria *id\_user*.
- **Tabla *offer*:** Incluye todos los datos de las ofertas lanzadas por los comercios, y la clave primaria *id\_offer*.
- **Tabla *rating\_business*:** Almacena las valoraciones de los comercios realizadas por los usuarios de la aplicación Android, junto con el identificador de comercio y el de usuario. A estos dos identificadores se les denomina *claves foráneas*<sup>6</sup>, ya que son iguales que la clave primaria de otra tabla y sirven para relacionar dichas tablas.
- **Tabla *business\_offers*:** Contiene la cantidad de unidades solicitadas y vendidas de cada oferta publicada por cada comercio. En esta tabla, *id\_offer* e *id\_bus* son

---

<sup>5</sup> Una llave o clave primaria es un indicador único en una tabla, por ejemplo un identificador.

<sup>6</sup> Una clave foránea se define como un campo de una tabla que hace referencia al campo de una clave principal de otra tabla, e indica cómo están relacionadas las tablas. Los datos en ambos campos deben coincidir, aunque los nombres de los campos no sean los mismos.

claves foráneas.

- **Tabla *requested\_offers*:** Guarda las ofertas solicitadas por cada usuario junto a la fecha de solicitud y un campo que nos dice si esa oferta ha sido comprada o no. *Id\_user* e *id\_offer* son claves foráneas.

La estructura final de la base de datos descrita, junto con las relaciones entre sus tablas, se puede observar en la Figura 2.4

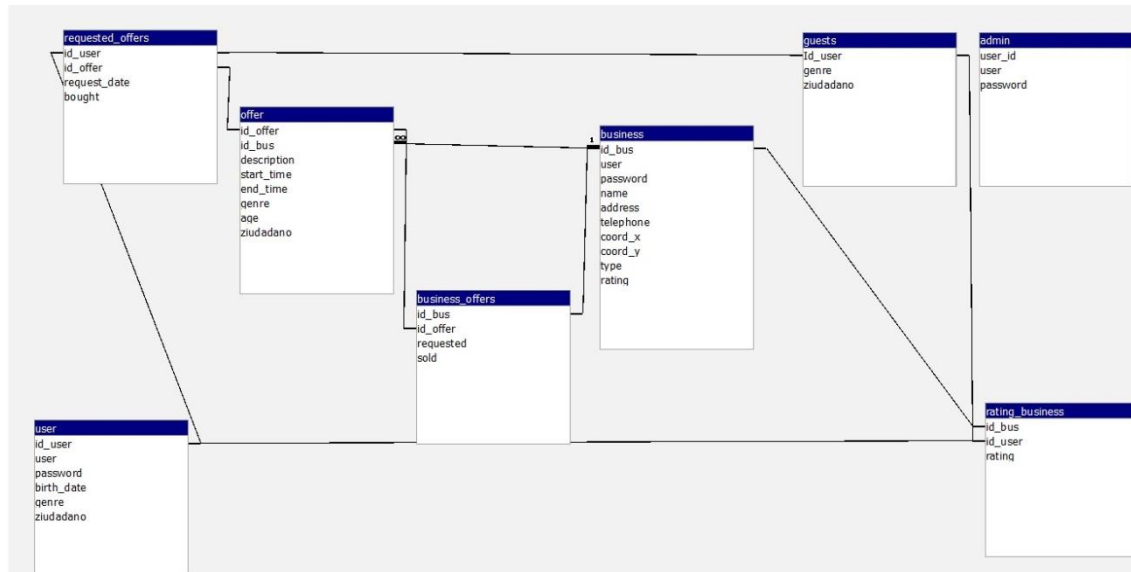


Figura 2.4. Estructura general de la base de datos.

## 2.2 Aplicación web

Se ha desarrollado una aplicación web eficiente, sencilla e intuitiva, con el objetivo de que los comerciantes no tengan dificultades al utilizarla, y evitar la necesidad de impartir cursos de formación. El objetivo es permitir a los comercios publicar sus ofertas en el sistema de una manera sencilla, así como mostrarles dichas ofertas junto con sus solicitudes, además de un historial de las ofertas antiguas.

No obstante, además de este modo de uso, que denominaremos modo *comercio*, se ha decidido crear un modo *administrador*, encargado de facilitar la administración de cuentas de usuario del sistema completo y al que únicamente tiene acceso el administrador del sistema. Para entender de una manera más visual este apartado, se puede observar la parte de la Figura 2.1 correspondiente a la aplicación web.

Se ha utilizado un método básico, que se podría aplicar a cualquier proyecto de desarrollo de aplicaciones, el cual se describe en 2.2.2. Se ha seleccionado este método

debido a su simplicidad de ejecución y sus excelentes resultados. A continuación se describe cómo se ha llevado a cabo el desarrollo completo de la aplicación web.

### 2.2.1 Diseño de la estructura de la aplicación web

Esta sección describe el diseño de la estructura utilizada en la aplicación web, tanto a nivel de sistema de archivos como a nivel de ventanas de navegación.

En la Figura 2.5 se puede observar la estructura fundamental de la aplicación web así como la navegación entre las distintas ventanas de las que se compone.

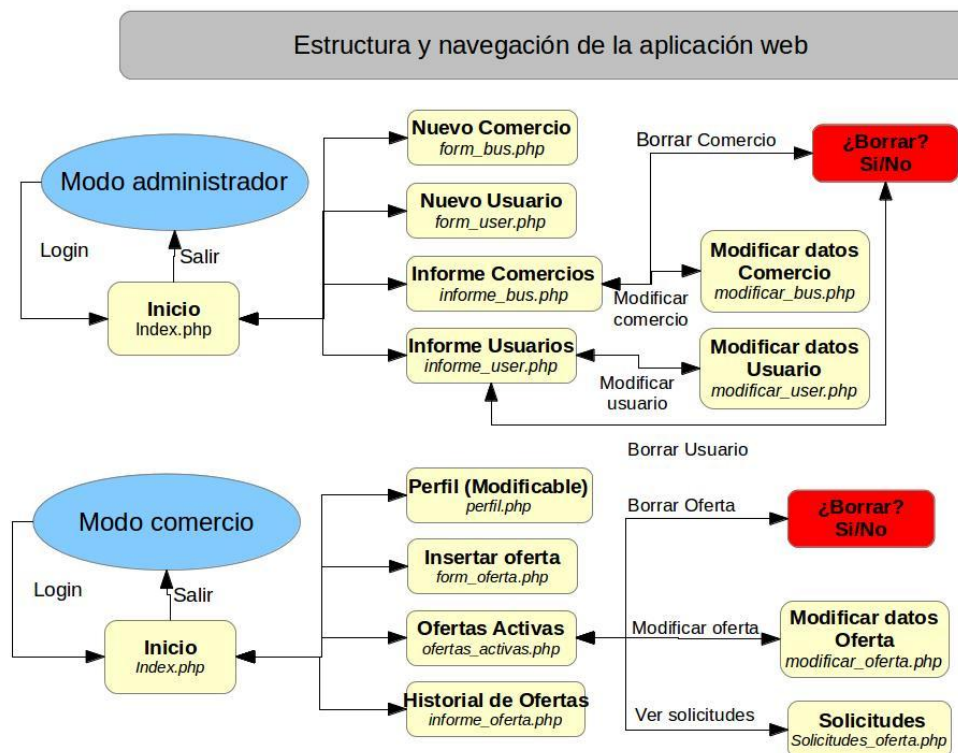


Figura 2.5. Estructura y diagrama de navegación de la aplicación web.

#### 2.2.1.1 Arquitectura de la aplicación web

En este apartado se va a explicar cómo se distribuye el sistema de archivos en la máquina virtual desplegada en la nube que soporta la aplicación web.

Todos los archivos que forman la aplicación web se encuentran almacenados en un directorio en la misma máquina virtual donde se encuentra el servidor principal.

La aplicación web consta de:

- Un archivo *index.php* donde se encuentra programada la ventana de inicio
- Una carpeta llamada *data* que contiene el resto de ventanas de la aplicación, es

decir, todos los archivos PHP que definen el resto de la aplicación.

- Una carpeta llamada *css* que contiene los estilos utilizados en el diseño de los componentes utilizados en la web.
- Una carpeta llamada *images*, donde están las imágenes utilizadas en la aplicación.
- Una carpeta llamada *admin*, que contiene el modo administrador, es decir dentro de esta carpeta hay otra estructura exactamente igual, archivo *index.php* y carpetas *data*, *css* e *images*.

En la Figura 2.6 se puede observar el aspecto de la estructura de carpetas dentro de la máquina virtual

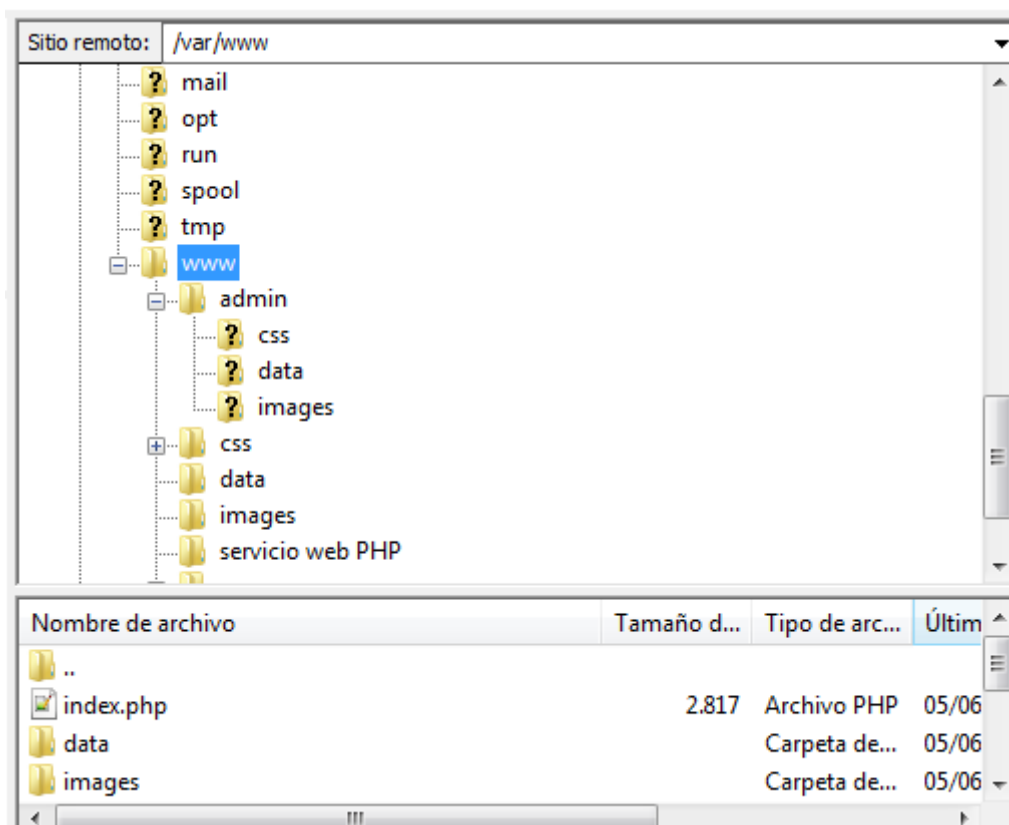


Figura 2.6. Estructura de carpetas de la aplicación web.

#### 2.2.1.2 Análisis de la aplicación, modelo de datos y navegación entre ventanas

Dependiendo del modo de uso de la aplicación web se van a utilizar unos datos u otros. En el caso del modo *administrador* se han definido cuatro funciones esenciales:

- **Nuevo comercio:** Cuando un comercio quiere formar parte del sistema deberá facilitar su información (nombre de la empresa, dirección, teléfono, etc.). Desde esta ventana el

- **Informe de usuarios:** La idea es la misma que para los comercios, es decir, se muestran los usuarios del sistema junto con su perfil, además de incorporar la opción de borrar o modificar sus datos.

En la Figura 2.7 se muestran respectivamente los conjuntos de datos utilizados en el modo *administrador*, a) datos de usuarios y b) datos de comercios. También muestra los datos de las ofertas que serán empleados en el modo *comercio*, c) Datos de ofertas

Datos de usuarios	Datos de comercios	Datos de ofertas
Identificador de usuario	Identificador de comercio	Identificador de oferta
Correo electrónico	Nombre usuario	Descripción
Contraseña de acceso	Contraseña de acceso	Fecha inicio
Género	Nombre del comercio	Fecha límite
Fecha de nacimiento	Dirección	Género al que va dirigida
Usuario tarjeta ciudadana	Teléfono	Edad a la que va dirigida
	Coordenada latitud	¿Para usuario de tarjeta ciudadana?
	Coordenada longitud	Número de solicitudes
	Tipo de comercio	Número de compras

Figura 2.7. Datos utilizados en la aplicación web: a) Datos de usuarios b) Datos de comercios c) Datos de ofertas.

En el caso del modo *comercio* se han implementado tres funciones principales:

- **Insertar oferta:** Consiste en un formulario en el que introducir la información de cada oferta para realizar la correspondiente publicación.
- **Ofertas activas:** Muestra en una tabla las ofertas activas del comercio y la información acerca del número de solicitudes y compras. Además incluye la opción de



borrar o modificar dichas ofertas y la opción de ver las solicitudes enviadas por los clientes, así como la posibilidad de registrar esas solicitudes como *oferta vendida*.

- **Historial de ofertas:** Muestra el historial de ofertas junto con estadísticas de número de solicitudes y número de compras de cada oferta, permitiendo saber cuáles han sido las más exitosas.

En los tres casos se manejan los datos de las ofertas, representados en la Figura 2.7, c).

Este modo también incluye un acceso al perfil del comercio, teniendo la opción de modificarlo si fuera necesario. Esta función maneja datos de información del comercio, los mismos que en el caso del modo administrador (Figura 2.7, b).

Como apoyo a este apartado y en general a todo el Apartado 2.2, se ha facilitado un manual de usuario de la aplicación web, disponible en el Anexo B “Manual de usuario de la aplicación web”.

### **2.2.2 Metodología de desarrollo de la aplicación web**

Esta aplicación se ha realizado en paralelo con el resto de los componentes del sistema, es decir, a pesar de haber partido de una idea clara y de unos prototipos de ventana concretos, estos se han ido modificando a lo largo del proyecto debido a las nuevas ideas, o a las modificaciones que se han realizado en la base de datos.

Se ha utilizado la misma metodología a lo largo de todo su desarrollo, que se ha basado en la iteración de las siguientes fases:

- Diseño de los elementos necesarios. Diseño de las ventanas del navegador.
- Análisis, obtención y presentación de la información. Se analiza qué información se debe mostrar y cómo obtenerla.
- Interacción (botones, cuadros de texto, etc.). Se configura la interacción de los distintos elementos que componen las ventanas del navegador.
- Pruebas. Se realizan las pruebas pertinentes para comprobar el correcto funcionamiento de todos los elementos, ya sea por separado o en conjunto.

Esta metodología empleada en el desarrollo de la aplicación web se describe más en detalle en el Anexo J “Metodología de desarrollo de la aplicación web”.

## 2.2.3 Decisiones de diseño adoptadas

### 2.2.3.1 Representación de la información

Teniendo en cuenta que la aplicación será utilizada por un público con un amplio rango de edades, uno de los principales objetivos del proyecto ha sido conseguir una aplicación fácil de usar, intuitiva y que muestre toda la información al usuario con pocos *clicks*.

Por ello se ha desarrollado una interfaz visual simple, con botones grandes y llamativos. Además, la forma elegida para representar la información ha sido mediante tablas, pues se ha pensado que es la mejor manera de estructurar la información referente tanto a ofertas publicadas, en el caso del modo *comercio*, como a usuarios o comercios en el modo *administrador*. Esta idea se puede ver reflejada en las Figuras 2.8 y 2.9 que muestran la pantalla principal del modo *comercio* y la representación de la información referente al historial de ofertas de una empresa, respectivamente.



Figura 2.8. Pantalla de inicio del modo comercio.

Id de oferta	Descripción	Hora de Inicio	Hora Limite	Género	Edad	Ziudadano	Nº de solicitudes	Nº de compras
8	2x1 en cafés	2014-06-05 08:00:00	2014-06-05 11:00:00	Ambos	Todo público	N	4	4
9	10 % descuento en bono comidas	2014-06-06 09:00:00	2014-06-06 14:00:00	Ambos	Todo público	N	4	2
10	Con un café, bollo gratis	2014-06-09 08:00:00	2014-06-09 13:00:00	Ambos	Todo público	Y	2	1
11	3x2 en cervezas	2014-06-09 12:00:00	2014-06-09 14:00:00	Ambos	Mayor de 18	N	4	2
12	Compra un bono comida y come hoy gratis!!	2014-06-04 13:00:00	2014-06-04 19:00:00	Ambos	Todo público	N	5	4

Número: 5

Figura 2.9. Ventana del historial de ofertas del modo comercio.

### 2.2.3.2 Comunicación con la base de datos del sistema

La comunicación con la base de datos del sistema es relativamente sencilla, ya que al tratarse de una aplicación basada en HTML5 y PHP, basta con incluir un archivo en el que se configure la conexión, el archivo *conexion.php* cuyo contenido es:

```
<?php
// Conexión a la base de datos MySQL
// Definición de Los parámetros de conexión
define('DB_SERVER','localhost'); // Dirección del servidor
define('DB_NAME','mycity4me'); // Nombre de la base de datos
define('DB_USER','root'); // Usuario de acceso
define('DB_PASS','toor'); // Contraseña de acceso
// Conexión pasando Los parámetros definidos anteriormente
$con = mysql_connect(DB_SERVER,DB_USER,DB_PASS);
mysql_select_db(DB_NAME,$con);
?>
```

Posteriormente, en el resto de archivos encargados de definir las ventanas de la aplicación basta con añadir la siguiente línea de código para que la conexión sea establecida:

```
require "data/conexion.php";
```

## 2.3 Aplicación Android

En este apartado se va a describir la aplicación Android. Para la explicación nos referiremos al esquema global presentado en la Figura 2.1.

A continuación se muestra un breve estudio sobre algunas aplicaciones ubicadas en el mismo ámbito que *MyCity4Me*. Más tarde se explica cada una de las partes de las que consta el desarrollo de esta aplicación, es decir *análisis*, *diseño* e *implementación*, incluyendo la metodología que se ha aplicado para llegar al desarrollo de *MyCity4Me*.

### 2.3.1 Aplicaciones relacionadas

Este apartado tiene como objetivo estudiar otras aplicaciones cuyo objetivo tenga cierta similitud con el de *MyCity4Me*.

#### 2.3.1.1 Foursquare

Una aplicación relacionada con *MyCity4Me* es *Foursquare*, descrita en [23], que muestra información al usuario y sus amigos en función de su localización. El usuario puede compartir y registrar dónde ha estado y obtener recomendaciones y ofertas en base a los lugares que tanto él como sus amigos han visitado. Permite introducir comentarios, y funciona de manera parecida a una red social. Esto ha planteado un problema a los desarrolladores de la aplicación, pues están compitiendo con *Facebook*, que además de ser la red social más de moda, está comenzando a incluir contenidos de este estilo, como enviar ofertas, anuncios, recomendaciones de amigos, basados en la localización. Además, está encaminada principalmente al público joven, a diferencia de la aplicación presentada en este PFC, que está pensada para un público general, gracias a su sencillez de uso. También hay que tener en cuenta que *Foursquare* no cuenta con un mapa donde ver todas las ofertas activas en un preciso instante, mientras que *MyCity4Me* sí lo incluye.

#### 2.3.1.2 El Tenedor

*El Tenedor*, aplicación descrita en [24], también tiene cierta semejanza con la aplicación desarrollada en este PFC, pues se trata de una aplicación que muestra información y ofertas de miles de restaurantes de toda España. Ahora bien, sólo se centra en restaurantes, a diferencia de *MyCity4Me*, que sirve para el pequeño comercio en general, incluyendo restaurantes.

Esta aplicación permite encontrar restaurantes y ofertas disponibles cerca del lugar donde estás. Permite hacer reservas inmediatas sin necesidad de prepagos, consultar una ficha detallada y también manda alertas si el restaurante de tu zona tiene alguna oferta nueva.

Lo que principalmente diferencia *MyCity4Me* de esta aplicación es que favorece al pequeño comercio en general, no sólo a los restaurantes.

#### 2.3.1.3 Mycoyote

*Mycoyote*, aplicación definida en [25], es una aplicación que hace uso de la herramienta de localización para avisar de la ubicación de radares, fijos, móviles, de semáforo y de tramo, además de ser un asistente en la carretera. Esta aplicación tiene cierta semejanza con *MyCity4Me*, al tratarse de una herramienta que trata con localizaciones geográficas.

\*\*\*

A diferencia de estas aplicaciones, *Mycity4Me*, al estar promovida por el Ayuntamiento, está pensada en principio para la ciudad de Zaragoza, lo cual no implica que no sea perfectamente escalable y replicable para su uso en otras ciudades.

### 2.3.2 **Análisis de la aplicación**

Como en cualquier proyecto de software, hay que determinar cuáles son los requisitos que debe satisfacer el sistema. A continuación se detallan los requisitos de la aplicación móvil, que definen las acciones fundamentales que deben ser llevadas a cabo al recibir información, procesarla y producir resultados:

- Funcionamiento como usuario o invitado.
- Opción de registro de nuevos usuarios.
- Visualización del perfil de usuario.
- Menú de ofertas activas ordenadas por hora límite más próxima y tipo de comercio, escogidas en función del perfil del usuario (género, edad, propietario de tarjeta ciudadana) y de su posición geográfica.
- Información acerca de cada oferta y comercio.
- Mapa con todos los comercios con ofertas activas geo-posicionados.
- Posición del usuario en el mapa.
- Presentación de información del comercio sobre el mapa.
- Acceso a ofertas lanzadas por los comercios que aparecen en el mapa.

- Posibilidad de valorar un comercio.
- Posibilidad de solicitar una oferta.
- Ruta desde donde está el usuario hasta establecimiento objetivo.
- Conexión con servicios web externos mediante protocolo HTML y formato de mensajes JSON (tratamiento de datos de la base de datos) o XML (envío de datos a *Orion*).

### 2.3.3 Diseño de la aplicación

Una vez definidos los requisitos de la aplicación, la siguiente tarea consiste en plantear su diseño. En primer lugar presentamos la arquitectura de la aplicación y a continuación se detalla el modelo de los datos utilizado.

#### 2.3.3.1 Arquitectura de la aplicación

La arquitectura de la aplicación se puede dividir en tres partes. La primera parte consiste en la aplicación propiamente dicha, es decir, código Java, clases, imágenes, archivos de configuración, etc.

La segunda parte está formada por las librerías externas a Android, añadidas al proyecto para aportar nuevas herramientas al entorno de desarrollo.

La tercera parte comprende las fuentes de datos de las que se nutre la aplicación. Existen dos tipos de fuentes de datos:

- Externas: Fuentes de datos ajenas al dispositivo móvil, normalmente accesibles mediante servicios web. En este caso el servicio web ha sido creado en este proyecto, mediante archivos PHP almacenados en la nube, en la misma máquina virtual que los archivos de la aplicación web y que la base de datos. Estos archivos permiten establecer la comunicación entre la aplicación Android y la base de datos del sistema, permitiendo así realizar consultas sobre los datos. Este sistema se describe más en detalle en el apartado 2.3.5.3.
- Internas: Se encuentran localizadas en el propio dispositivo móvil, en concreto en la tarjeta de memoria SD, donde se almacenan imágenes, vídeos, etc. También el fichero XML *sharedPreferences* sirve para almacenar datos privados en pares clave-valor, que posteriormente sirven, por ejemplo, para no tener que introducir los datos de acceso cada vez que se quiere entrar a una aplicación.

Esta breve descripción de la arquitectura se expone en la Figura 2.10.

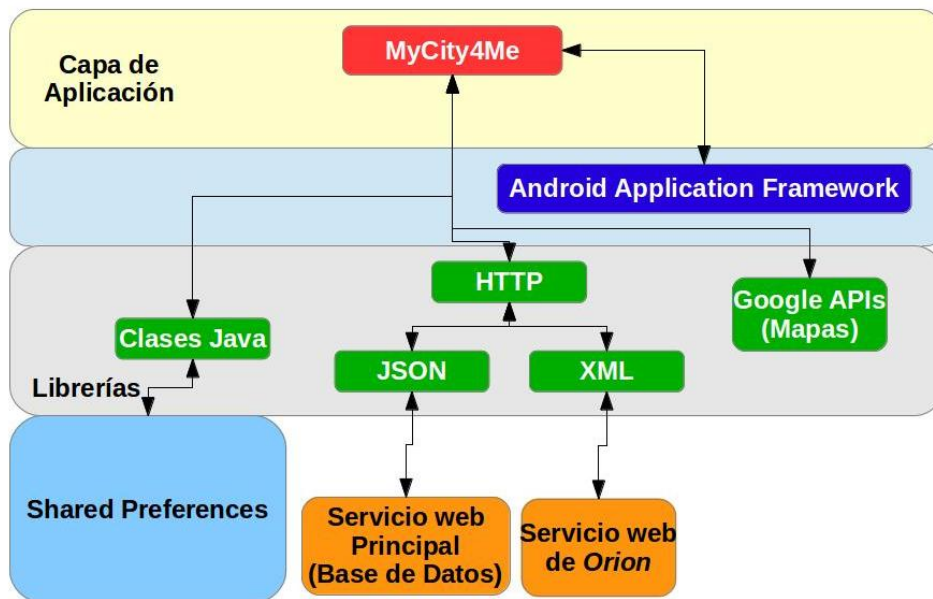


Figura 2.10. Arquitectura de la aplicación.

Las tres partes mencionadas al inicio de este apartado, se encuentran comunicadas entre sí. La comunicación entre la capa de aplicación y la de librerías se lleva acabo directamente, o mediante el *Application Framework*, zona del sistema operativo Android donde están las herramientas del desarrollador. Algunas herramientas incluidas en el *Application Framework*<sup>7</sup> y que han sido utilizadas en *MyCity4Me* son:

- **View System.** El sistema de *views* consiste en un conjunto de elementos que ayudan a construir las interfaces de usuario: botones, cuadros de texto, listas y elementos más avanzados como un navegador web o un visor de Google Maps. Esto se utiliza contantemente en *MyCity4Me* para construir las interfaces de usuario.
- **Activity manager.** El gestor de *Activities* se encarga de administrar la pila de actividades de la aplicación, así como su ciclo de vida. Este gestor tiene un funcionamiento por defecto, de manera que si se pasa de un *activity* a otro, el *activity* del que ha salido, se queda en la pila y si el usuario pulsa el botón “atrás”, la aplicación vuelve a dicho *activity*. En *MyCity4Me* hay algunos casos en los que tras acceder a un *activity*, no se quiere volver al anterior, sino a otro. Para conseguirlo hay que “destruir” ese *activity* al salir de él, escribiendo “*finish()*”.
- **Resource manager.** El gestor de recursos se encarga de gestionar todos los

<sup>7</sup> <http://developer.android.com/intl/es/guide/components/index.html>

elementos que forman parte de la aplicación y están fuera del código como cadenas de texto, imágenes, sonidos, *layout*, etc. Normalmente, los textos fijos de las aplicaciones se guardan en *String* en un archivo XML. En *MyCity4Me*, se han creado dos archivos de XML de *String*, uno con las cadenas de texto en Español, y otro en Inglés, permitiendo que en función de los ajustes del móvil se muestre el texto en el idioma correspondiente. Esto es posible gracias al gestor de recursos

- **Location manager.** El gestor de localización permite determinar la posición geográfica del dispositivo Android mediante GPS o mediante el análisis de las redes disponibles, así como trabajar con mapas. En *MiCity4Me* se utiliza para obtener la posición del usuario y enviar ofertas en función de ello, y también en el mapa, para localizar todos los comercios.
- **Window manager.** El gestor de ventanas se encarga de organizar lo que se muestra en pantalla. Se utiliza cada vez que se muestra una ventana.

La comunicación entre la aplicación y las fuentes de datos depende del tipo de fuente de información. En el caso de las fuentes externas, la comunicación se establece por medio de la red de datos 3G disponible en el móvil o Wi-Fi; en el caso de las fuentes internas, Android utiliza librerías para realizar dicha comunicación.

En el Anexo I.2 se explican más en detalle los métodos comunicación utilizados.

#### 2.3.3.2 Modelo de datos

Tras la descripción de la arquitectura de la aplicación y el análisis de los requisitos, se va a proceder a describir la organización de la información tratada en la aplicación, es decir, el modelo de datos.

El modelo de datos se va a apoyar en la base de datos creada en la infraestructura de datos en la nube, cuya descripción podemos encontrar en el Apartado 2.1.2.

Se dispone de tres objetos Java para manejar la información en la aplicación:

- Objeto *Profile*: se utiliza para manejar los datos del perfil de usuario.
- Objeto *Business*: se utiliza para manejar información referente a los comercios que se muestran en el mapa.
- Objeto *UserDeal*: se utiliza para manejar la información referente a las ofertas y mostrarla en el menú de ofertas.



La Figura 2.11 muestra la estructura de estos objetos.

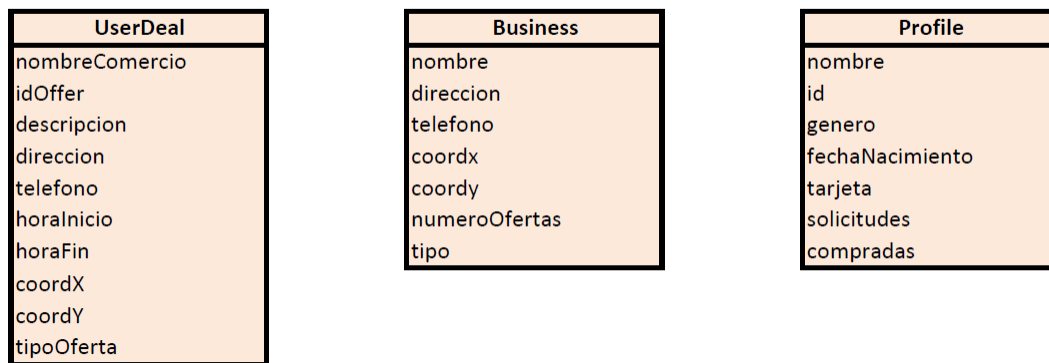


Figura 2.11. Objetos Java utilizados en la aplicación, junto con sus atributos.

Además de estos datos, existen otros que no se utilizan a través de objetos. Estos datos o bien se van pasando de un *Activity* a otro, a través de los *Intent* (ver la definición y más detalles en el Anexo F), para tenerlos disponibles a lo largo de toda la aplicación, como son los datos *user* e *iduser*, o bien se generan en la propia aplicación cuando son necesarios, como pueden ser las coordenadas de la posición actual del dispositivo móvil, o el estado del GPS (activado o desactivado).

### 2.3.4 Metodología de desarrollo de la aplicación

El desarrollo de la aplicación móvil ha sido la parte más costosa de este PFC en cuanto al tiempo empleado. Antes de entrar en la metodología utilizada en el desarrollo de *MyCity4Me*, hay que mencionar que la fase relacionada con el diseño gráfico de la aplicación se ha realizado en todo momento a la par que la fase de implementación. También se debe mencionar que, tal y como se ha explicado en el apartado referente a la aplicación web, las distintas partes del sistema se han desarrollado en paralelo, de manera que la aplicación Android ha ido evolucionando en función de modificaciones o actualizaciones del sistema completo.

Después de conocer lo básico acerca de los recursos gráficos proporcionados por el SDK<sup>8</sup> de Android, y tras diversos test realizados probando dichos recursos, se puede comenzar a diseñar un prototipo de interfaz gráfica. Android permite plasmar fácilmente un diseño previo, obtenido durante la fase de desarrollo, ya que cada una de las ventanas de navegación diseñadas se corresponde con una parte independiente de código fuente.

<sup>8</sup> <http://developer.android.com/intl/es/sdk/index.html>

Esto permite que el desarrollo de la aplicación sea sencillo y modular, y por ello, la metodología que se ha seguido se ha basado en la iteración de los siguientes pasos:

- Diseño gráfico de la ventana mediante la edición del archivo XML correspondiente, con la ayuda del asistente gráfico del entorno de programación.
- Análisis de la información necesaria en la ventana de navegación correspondiente, así como la obtención y presentación de la misma.
- Dotación de interacción con el usuario a todos los elementos presentes en la ventana (botones, cuadro de texto, etc.).
- Análisis “prueba y error” de cada componente.

Una descripción detallada de los pasos seguidos en el método de desarrollo se haya explicada en el Anexo I “Metodología de desarrollo de la aplicación Android”.

### 2.3.5 Decisiones de diseño adoptadas

A lo largo de este apartado, se van a describir los puntos críticos que han hecho tomar unas decisiones determinadas en cuanto al diseño de la aplicación.

#### 2.3.5.1 Representación de la información

Para desarrollar una aplicación intuitiva, con la que acceder a la información con pocos *clicks*, se ha pensado en el uso de un mapa, en el que aparezcan los comercios, que proporcione la mayor información posible, y en la presentación de un listado con las ofertas, filtradas en función del tipo de comercio y el perfil de usuario.

Para ello se han utilizado una serie de herramientas proporcionadas por Android, descritas a continuación:

- **Listas:** En Android, uno de los *widget* más potentes proporcionados son las listas. Son secuencias de ítems que tienen la posibilidad de mostrar diferentes *widget* embebidos, y que permiten un desplazamiento en vertical automático. En *MyCity4me* se utilizan para mostrar las ofertas, y al pulsar sobre una de ellas se muestra toda su información, tal y como podemos ver en la Figura 2.12.
- **Toasts:** Un *toast* es un mensaje que se muestra en pantalla durante unos segundos al usuario para luego volver a desaparecer automáticamente sin requerir ningún tipo de actuación por su parte. Aparecen por defecto en la parte inferior de la pantalla, sobre un rectángulo gris ligeramente translúcido

y se suelen utilizar para mostrar información puntual. En esta aplicación se utilizan por ejemplo para señalar al usuario que no tiene el GPS activado, o que la solicitud de oferta se ha realizado con éxito (Figura 2.13).

- **Mapas:** Aunque existen mapas basados en código abierto (mapas OSM<sup>9</sup>), en este proyecto se ha optado por utilizar la API ofrecida por Google Maps, mucho más completa y difundida. Esta parte se explica más en detalle en el Apartado 2.3.5.2. El mapa puede verse en la Figura 2.16.
- **Diálogos:** Un diálogo es una pequeña ventana que solicita al usuario que tome una decisión o que introduzca información adicional. En esta aplicación se utiliza para preguntar al usuario si realmente quiere cerrar sesión cuando este pulsa el botón de cerrar sesión (ver Figura 2.14).
- **Barra de acción:** Se trata de una barra superior donde se muestra el título de la aplicación junto a un icono y un número limitado de opciones diferentes. En *MyCity4Me* esta barra incluye un acceso al perfil de usuario, un acceso al mapa de ofertas y un botón de cerrar sesión, dependiendo de la ventana en la que se encuentre el usuario, tal y como se aprecia en las Figuras 2.13 y 2.14.

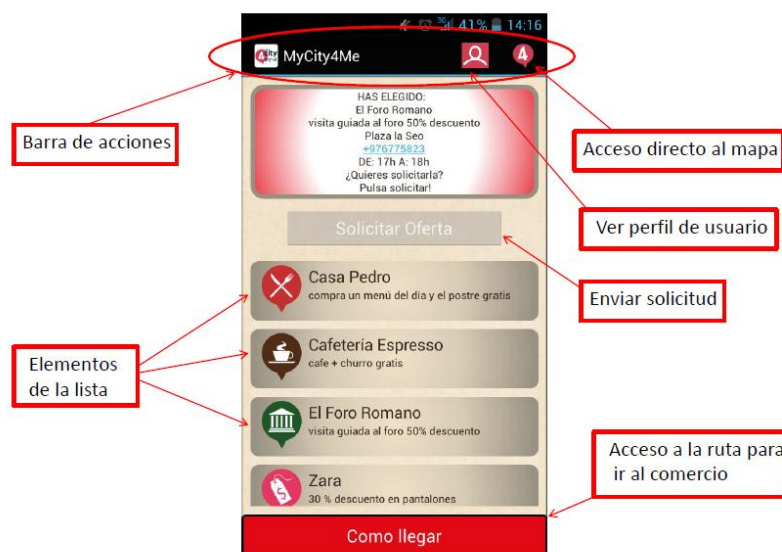


Figura 2.12. Ejemplo de una de una lista y la barra de acciones en MyCity4Me.

<sup>9</sup> Open Street Map, <http://www.openstreetmap.es/>



*Figura 2.13. Ejemplo de uso de los Toast y la barra de acciones en MyCity4Me.*



*Figura 2.14. Ejemplo de uso de un Dialog en MyCity4Me.*

### 2.3.5.2 Mapas utilizados

A continuación se explica el visor de mapas utilizado, así como toda la funcionalidad que este otorga a la aplicación. Como ya se ha comentado en el apartado anterior, hay diversas opciones. En concreto las más atractivas pueden ser los mapas de Google y los mapas de la librería OSM, estos últimos basados en código abierto. Se ha seleccionado la opción de Google dada su gran popularidad, ya que un porcentaje muy alto de la población los ha usado alguna vez y por tanto sabe utilizarlos e interpretarlos. Además son unos mapas muy completos.

El código necesario para activar los mapas en esta aplicación se encuentra detallado en el Anexo H.

El mapa se va a utilizar para mostrar todos los comercios que tienen ofertas activas en el momento en que el usuario entra en la aplicación, y permite el acceso a la lista de ofertas del comercio seleccionado. También se utiliza la aplicación *Navigator* de Google para el cálculo de rutas desde la posición del usuario hasta el comercio cuya oferta ha sido seleccionada.

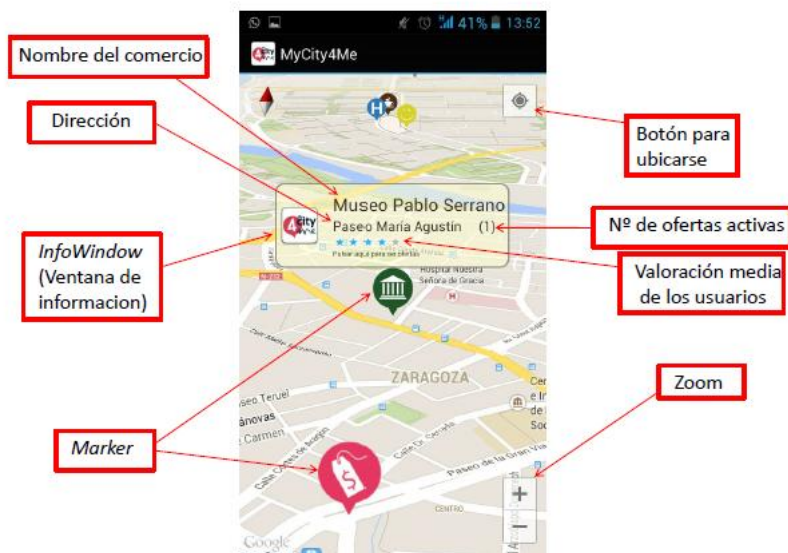
Como fase de personalización del mapa se han añadido los siguientes elementos:

- *Marker* personalizados para los distintos tipos de comercios.
- Botón de aumento y reducción del zoom (que indica el nivel en el que está).
- Botón que localiza al usuario en el mapa.
- Ventanas de información acerca del comercio seleccionado.

En resumen, el mapa muestra los comercios mediante el uso de *marker*, diseñados en función del tipo de comercio (Figura 2.15). A su vez, cuando el usuario pulsa uno de los *marker*, aparece una ventana de información, conocida como *infoWindow*, que contiene información detallada acerca del punto de interés seleccionado. En este proyecto, dicha ventana muestra el nombre del comercio, su dirección y el número de ofertas, así como una valoración calculada a partir de las valoraciones enviadas por los usuarios. En la Figura 2.16 se puede ver una captura del mapa presentado por *MyCity4Me*. Estas ventanas de información también son “*clickables*”, de manera que si un usuario pulsa una de ellas, accede al listado de ofertas activas del comercio. No obstante, la navegación en el mapa se describe más ampliamente en el Anexo A “Manual de usuario de la aplicación Android”.



*Figura 2.15. Iconos personalizados utilizados en MyCity4Me (Markers).*



*Figura 2.16. Captura del mapa modificado de MyCity4Me.*

### 2.3.5.3 Comunicación con la base de datos del sistema

En primer lugar se establecen los requisitos que debe cumplir la comunicación a través del servicio web PHP, cuya implementación esta descrita en el Anexo E “Implementación del servicio web PHP”.

Se pretende que la comunicación sea sencilla, y que el formato de los mensajes sea ligero para dar fluidez al sistema y que sean fáciles de tratar con código Java.

Pensando en estos requisitos mínimos, se ha llegado a la conclusión de que el mejor protocolo a usar es HTTP<sup>10</sup> mediante el uso de mensajes en formato JSON, es decir mensajes tipo clave-valor.

El motivo por el que se ha preferido el uso de HTTP, es que el servicio web ha sido desarrollado al mismo tiempo que el resto del sistema, y éste ha sido implementado mediante archivos PHP. Tanto PHP como el SDK de Android tienen soporte para este protocolo (HTTP).

En la Figura 2.17 se puede ver de forma sencilla en qué consiste lo descrito en este apartado.



*Figura 2.17. Sistema de comunicación entre Android y la base de datos.*

Por último se describen los distintos envíos y consultas que se realizan desde *MyCity4Me* al servicio web utilizando el protocolo HTTP y el formato de mensajes JSON:

- Nuevo registro, invitado o usuario personal.
- Verificación *Login/Password*.
- Consulta del perfil de usuario.
- Consulta de todos los comercios con ofertas activas para mostrarlos en el mapa.
- Consulta de todas las ofertas en función del perfil de usuario y del tipo de oferta

---

<sup>10</sup> Hypertext Transfer Protocol. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

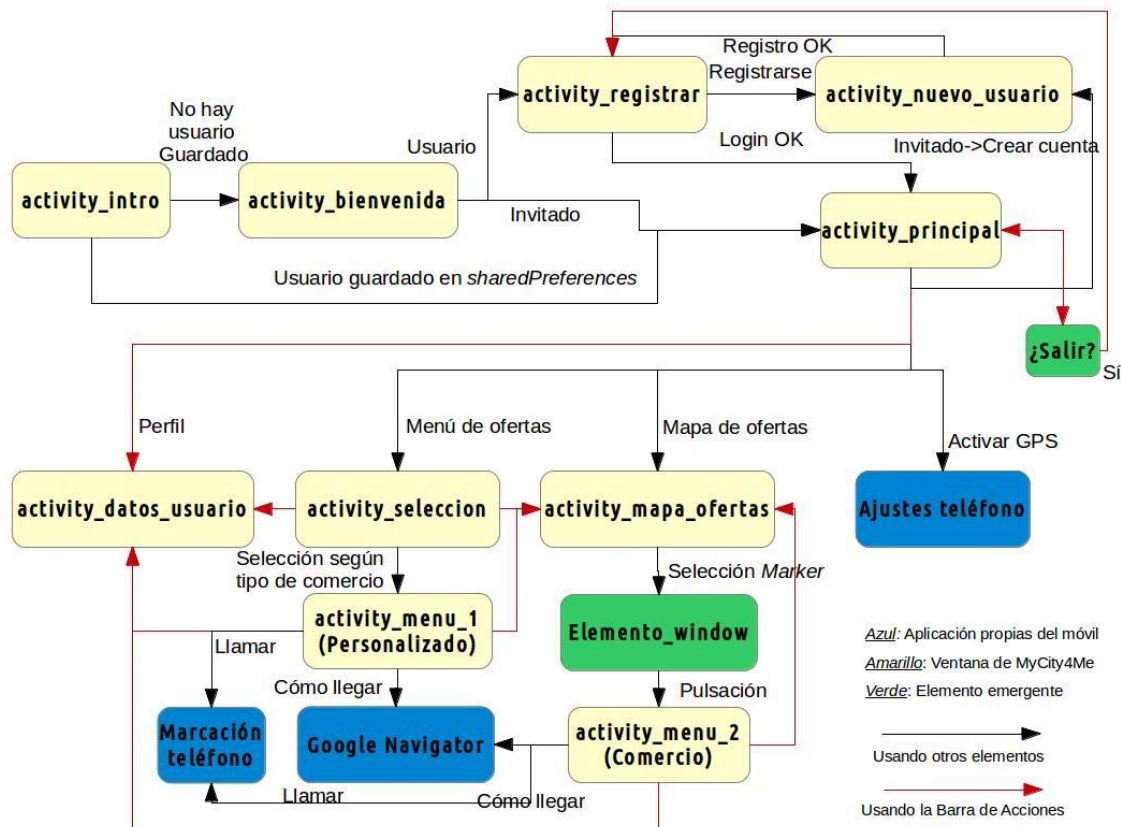
seleccionado.

- Consulta de todas las ofertas activas de un determinado comercio.
- Envío de solicitud de la oferta solicitada por el usuario.
- Envío de la valoración de un usuario de un comercio determinado.

### Consulta de la valoración global de los comercios.

### 2.3.6 Estructura de la aplicación

En este último apartado se va a describir brevemente la estructura de la aplicación móvil, cuyo esquema general se encuentra representado en la Figura 2.18. No obstante, esta parte se encuentra descrita visualmente, por medio de capturas de pantalla, en el Anexo A “Manual de usuario de la aplicación Android”.



*Figura 2.18. Estructura y navegación de la aplicación móvil MyCity4Me.*

A continuación, con ayuda de la Figura 2.18, se describen las funciones de las distintas ventanas de la aplicación:

- **Activity\_intro:** Se encarga de cargar los datos de usuario almacenados en el fichero *sharedPreferences* de manera que si no existen pasa a *activity\_bienvenida*; si existen, inicia la sesión y pasa directamente a



*activity\_principal*.

- **Activity\_bienvenida:** Permite elegir el modo de acceso a la aplicación. El modo *Invitado* genera un usuario de tipo invitado y pasa a *activity\_principal* y el modo usuario registrado pasa a la ventana de login, *activity\_registrar*.
- **Activity\_registrar:** Se encarga de tramitar el *login* a la aplicación. Además permite crear un nuevo usuario para lo que accede a *activity\_nuevo\_usuario*.
- **Activity\_nuevo\_usuario:** Permite introducir los datos del usuario a registrar y enviarlos a la base de datos. Después vuelve a *activity\_registrar*.
- **Activity\_principal:** Este es el eje de la aplicación, a partir de esta ventana se puede acceder al menú de selección de ofertas, *activity\_seleccion*, al mapa, *activity\_mapa\_ofertas* y al perfil, *activity\_datos\_usuario*. Además permite acceder a los ajustes del teléfono y en caso de ser un usuario registrado, permite cerrar la sesión.
- **Activity\_datos\_usuario:** Muestra los datos del usuario y mediante la barra de acciones, dispone de acceso directo al mapa de ofertas.
- **Activity\_seleccion:** Permite elegir el tipo de ofertas que quieres ver, pasando a *activity\_menu\_1*. Contiene acceso directo al mapa y al perfil a través de la barra de acciones.
- **Activity\_mapa\_ofertas:** Muestra la posición de los comercios con ofertas activas. Pulsando en los iconos, se muestra una ventana de información (*infoWindow*) con información del comercio. Esta ventana también se puede pulsar accediendo así al menú de ofertas activas de ese comercio, *activity\_menu\_2*.
- **Activity\_menu\_1:** Muestra una lista con las ofertas activas en función del tipo de oferta, al pulsar un elemento de la lista aparece una ventana con información sobre la oferta. Permite solicitar ofertas e iniciar el navegador para ver la ruta desde la posición del usuario hasta el comercio que ha lanzado la oferta.
- **Activity\_menu\_2:** Muestra una lista con las ofertas activas publicadas por el comercio seleccionado. Permite solicitar ofertas, valorar el comercio e iniciar el navegador.

Ambos menús tienen acceso directo al perfil y el mapa a través de la barra de acciones y pulsando sobre el teléfono que aparece junto con la información de la oferta permite iniciar el terminal de llamadas del teléfono con dicho teléfono marcado, facilitando la tarea si se quiere contactar con el comercio.



### **3 Conclusiones y líneas futuras**

En este capítulo final de la Memoria se va a valorar el cumplimiento de los objetivos a la vista de los resultados obtenidos, se van a indicar las posibles líneas futuras como continuación del presente PFC, y se valorará la experiencia conseguida con su realización.

#### **3.1 Conclusiones**

El resultado más apreciable es el correcto funcionamiento del sistema completo desarrollado a lo largo de este proyecto.

Además se puede afirmar el cumplimiento de todos los objetivos marcados al inicio de éste, sobre todo los dos principales, marcados por el Servicio de Ciudad Inteligente del Ayuntamiento de Zaragoza: utilizar la novedosa plataforma FI-WARE y desarrollar un sistema que sea capaz de aportar un beneficio al pequeño comercio. El sistema creado es completamente funcional y cumple correctamente los objetivos.

Del desarrollo de este proyecto se han adquirido una serie de experiencias interesantes, que se describen a continuación:

- El hecho de usar una infraestructura en la nube permite trabajar desde cualquier lugar y desde cualquier terminal.
- El conjunto de herramientas y recursos ofrecidos por la plataforma FI-WARE, ha permitido trabajar cómodamente en la nube, aunque al tratarse de un proyecto en desarrollo, todavía quedan pendientes muchas mejoras y mucha labor de documentación.
- El uso de JSON supone una ventaja respecto a XML, en este caso ha servido para dotar al proyecto de rapidez y sencillez en la transmisión de datos entre las distintas partes del sistema, lo que es muy importante dadas las exigencias de los usuarios.
- Android ha demostrado su potencial como plataforma de desarrollo de aplicaciones. Además tiene a disposición de los desarrolladores gran cantidad de información como manuales, tutoriales, etc. El entorno de programación es intuitivo y sencillo de manejar, lo que permite

programar con relativa rapidez. La popularidad de este sistema operativo permite que un gran número de usuarios tenga la posibilidad de usar la aplicación.

- El uso de los mapas de Google ha sido clave en *MyCity4Me*, ya que dota a la aplicación de diseño y sencillez.

### **3.2 Líneas futuras**

Las posibilidades de futuras líneas de trabajo son diversas y se enumeran a continuación:

- Introducción de la *Tarjeta Ciudadana* de Zaragoza como medio de confirmación de solicitudes de ofertas en los comercios o para otras funciones.
- Creación de un sistema que pueda sacar partido a los datos generados por la aplicación, llegando incluso a generar estudios de mercado. Por ejemplo, gracias a los datos almacenados en *Cosmos*, se podría elaborar un mapa con las zonas más concurridas por usuarios de *MyCity4Me*, lo que permitiría a los comercios centrar su publicidad en esas zonas.
- Mejora de la aplicación web añadiendo nuevas mejoras que la hagan aún más intuitiva y sencilla.
- Dotar de seguridad al sistema, por ejemplo, integrando sistemas de cifrado para proteger los datos.
- Mejoras sobre la aplicación Android, como nuevas formas de visualizar las ofertas, introducción de fotos de los comercios para facilitar su reconocimiento en la calle, etc.

### **3.3 Valoración personal**

Este proyecto ha tenido una inmensa cantidad de repercusiones sobre mí, sobre todo en sentido profesional. Todo lo que he aprendido a lo largo de este proyecto era inimaginable antes de empezarlo. El hecho de trabajar con tan gran variedad de tecnologías me ha abierto puertas que no sabía que existían.

Trabajar con la plataforma FI-WARE me ha permitido conocer y trabajar junto a profesionales del sector. La elección de este proyecto como uno de los finalistas del reto *Smart Business & Industry* de FI-WARE me ha otorgado una experiencia muy interesante y ha sido clave para la consecución de todos los objetivos marcados.

Además de todo esto, aprender a desarrollar aplicaciones Android, sistema operativo en pleno auge, y aplicaciones web, me ha abierto un camino inesperado.

Pensar que he realizado un proyecto completamente funcional, para la ciudad en que vivo y que puede beneficiar a la gente de la ciudad, hace que todo el desarrollo lo haya llevado a cabo con gran energía.

Este proyecto ha hecho que se activen mis ganas de aprender todo lo posible acerca de las tecnologías punteras en la actualidad.



## REFERENCIAS

- [1] FI-PPP, *Future Internet Public-Private-Partnership*. Disponible online: [www.fi-ppp.eu](http://www.fi-ppp.eu) [visitado diciembre 2013].
- [2] Denis Havlik, Sven Schade, Wout van Wijk, Thomas Usländer, Juanjo Hierro: Leveraging the Future Internet for the Environmental Usage Area, EnviroInfo 2011.
- [3] Juniper Research Whitepaper, “Consumer Cloud ~ There’s No Limit”, Mayo 2014. Disponible online en: [http://www.juniperresearch.com/shop/download\\_whitepaper.php?whitepaper=261](http://www.juniperresearch.com/shop/download_whitepaper.php?whitepaper=261) [visitado mayo 2014].
- [4] FI-WARE, proyecto principal del programa FI-PPP. Disponible online: [www.fi-ware.org](http://www.fi-ware.org) [visitado diciembre 2013].
- [5] Página web de acceso a FI-Lab: FI-LAB, *the open innovation Lab*. Disponible online: <https://account.lab.fi-ware.org/> [visitado enero 2014].
- [6] FI-WARE Glossary Global, definición de Generic Enabler, accesible en: [http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE\\_Glossary.Global](http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_Glossary.Global), [visitado junio 2014]
- [7] *Orion Context Broker*. Disponible online: <http://catalogue.fi-ware.org/enablers/publishsubscribe-context-broker-orion-context-broker> [visitado enero 2014].
- [8] *Cosmos - BigData Analysis*. Disponible online: <http://catalogue.fi-ware.org/enablers/bigdata-analysis-cosmos> [visitado enero 2014].
- [9] Máquina virtual Dalvik, *Dalvik Technical Information*, Disponible online: <http://source.android.com/devices/tech/dalvik/> [visitado enero 2014].
- [10] Actualización porcentajes de uso de las distintas versiones de Android. Disponible online: <http://appskyworld.com/2014/01/google-actualiza-los-porcentajes-de-uso-de-las-diferentes-versiones-de-android-en-la-pagina-de-desarrolladores/> [visitado febrero 2014].
- [11] Porcentajes de uso de los distintos sistemas operativos para móviles. Disponible online: <http://blog.uchceu.es/informatica/ranking-de-sistemas-operativos-mas-usados-para-2014/> [visitado febrero 2014].
- [12] Comparacion JSON y XML. Disponible online: <http://www.json.org/xml.html> [visitado febrero 2014].
- [13] Instituto Nacional de Estadística, “Índices de Comercio al por Menor (ICM). Base 2010, Abril 2014”. Datos provisionales. Disponible online: <http://www.ine.es/daco/daco42/daco4215/ccm0414.pdf> [visitado enero 2014].
- [14] Comscore, “EU5 Smartphone Penetration Reaches 55 Percent in October 2012”, Dec 2012. Disponible online [http://www.comscore.com/Insights/Press\\_Releases/2012/12/EU5\\_Smartphone\\_Penetration\\_Reaches\\_55\\_Percent\\_in\\_October\\_2012](http://www.comscore.com/Insights/Press_Releases/2012/12/EU5_Smartphone_Penetration_Reaches_55_Percent_in_October_2012) [visitado abril 2014].

- [15] Ideas selected to compete in the FI-WARE Challenges: Part I – Smart Business & Industry. Disponible online: <http://www.fi-ware.org/2014/01/15/fi-ware-challenges-smart-business-finalists/> [visitado abril 2014].
- [16] Louise Barkhuus, Context Information vs. Sensor Information: A Model for Categorizing Context in Context-Aware Mobile Computing. SIMULATION SERIES, 2003, vol. 35, no 1, p. 127-133. Disponible online: [http://www.itu.dk/people/barkhuus/barkhuus\\_cst.pdf](http://www.itu.dk/people/barkhuus/barkhuus_cst.pdf) [Visitado enero 2014].
- [17] *What is MapReduce?* Disponible online en: <http://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/> [Visitado en enero 2014].
- [18] HDFS Architecture Guide. Disponible online en: [http://hadoop.apache.org/docs/r1.2.1/hdfs\\_design.html#Introduction](http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html#Introduction) [Visitado en enero 2014].
- [19] *Cygnus connector*. Disponible en: <https://github.com/telefonicaid/fiware-connectors/blob/release/0.1/flume/README.md> [visitado mayo 2014].
- [20] *Welcome to Apache Flume*. Disponible online en: <http://flume.apache.org/> [Visitado en mayo 2014].
- [21] Reglas de normalización de bases de datos. Disponible online: <http://www.eet2mdp.edu.ar/alumnos/MATERIAL/MATERIAL/info/infonorma.pdf> [visitado febrero 2014].
- [22] *Tarjeta Ciudadana* de Zaragoza. Disponible online en: [http://www.zaragoza.es/ciudad/sectores/tarjetaciudadana/detalle\\_Tramite?id=22960](http://www.zaragoza.es/ciudad/sectores/tarjetaciudadana/detalle_Tramite?id=22960) [Visitado en diciembre 2013].
- [23] *Aplicación Foursquare*. Disponible online en: <https://play.google.com/store/apps/details?id=com.joelapenna.foursquared&hl=es> [Visitado en enero 2014].
- [24] *Aplicación El Tenedor*. Disponible online en: <https://play.google.com/store/apps/details?id=com.lafourchette.lafourchette&hl=es> [Visitado en enero 2014].
- [25] *Aplicación Mycoyote*. Disponible online en: <http://www.mycoyote.es/es/productos/icoyoteapp> [Visitado en enero 2014]
- [26] Publish/Subscribe Broker - Orion Context Broker - User and Programmers Guide. Disponible en: [https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Publish/Subscribe\\_Broker - Orion Context Broker User and Programmers Guide#Entity Creation](https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Publish/Subscribe_Broker_-_Orion_Context_Broker_User_and_Programmers_Guide#Entity_Creation) [visitado enero 2014].



# ACRÓNIMOS

**API** Application Programming Interface

**CSS** Cascading Style Sheets

**FI** Future Internet

**FI-PPP** Future Internet Public-Private-Partnership

**GE** Generic Enabler

**GPS** Global Positioning System

**HDFS** Hadoop Distributed File System

**HTML** HyperText Markup Language

**HTTP** HyperText Transfer Protocol

**JSON** JavaScript Object Notation

**OSM** Open Street Maps

**PFC** Proyecto Fin de Carrera

**PHP** PHP Hypertext Pre-processor

**SDK** Software Development Kit

**SQL** Structured Query Language

**SVG** Scalable Vector Graphics

**TIC** Tecnologías de la Información y las Comunicaciones

**W3C** World Wide Web Consortium

**XHTML** eXtensible HyperText Markup Language

**XML** eXtensible Markup Language



## FIGURAS

1.1.	Estructura básica del sistema y las tecnologías utilizadas.	13
1.2.	Porcentaje de uso de las versiones Android. Enero 2014.	16
1.3.	Distribución del uso de sistemas operativos para dispositivos móviles.	17
2.1.	Estructura general del sistema.	21
2.2.	Ejemplo de información de contexto.	23
2.3.	Proceso de almacenamiento de coordenadas combinando el uso de <i>Orion</i> y <i>Cosmos</i> , a través del conector <i>Cygnus</i> .	24
2.4.	Estructura general de la base de datos	26
2.5.	Estructura y diagrama de navegación de la aplicación web.	28
2.6.	Estructura de carpetas de la aplicación web.	29
2.7.	Datos utilizados en la aplicación web. a) Datos de usuarios b) Datos de comercios c) Datos de ofertas.	30
2.8.	Pantalla de inicio del modo <i>comercio</i> .	32
2.9.	Ventana del historial de ofertas del modo <i>comercio</i> .	33
2.10.	Arquitectura de la aplicación.	37
2.11.	Objetos Java utilizados en la aplicación, junto con sus atributos.	39
2.12.	Ejemplo de una de una lista y la barra de acciones en <i>MyCity4Me</i> .	41
2.13.	Ejemplo de uso de los <i>Toast</i> y la barra de acciones en <i>MyCity4Me</i> .	42
2.14.	Ejemplo de uso de un <i>Dialog</i> en <i>MyCity4Me</i> .	42
2.15.	Iconos personalizados utilizados en <i>MyCity4Me</i> ( <i>Markers</i> ).	44
2.16.	Captura del mapa modificado de <i>MyCity4Me</i> .	44
2.17.	Sistema de comunicación entre Android y la base de datos.	45
2.18.	Estructura y navegación de la aplicación móvil <i>MyCity4Me</i> .	46
A.1.	Pantalla de inicio, selección de modo de acceso.	64
A.2.	Ventana principal en modo invitado.	64
A.3.	Ventana de registro.	65

A.4.	Ventana de <i>login</i> .	65
A.5.	Ventana principal en modo usuario registrado.	66
A.6.	Perfil de usuario.	67
A.7.	Filtrado de ofertas en función del tipo de comercio.	67
A.8.	Aspecto de la lista de ofertas después de pulsar el botón <i>Todas las ofertas</i> .	68
A.9.	Mapa de ofertas.	69
A.10.	Lista de ofertas del comercio seleccionado en el mapa.	70
B.1.	Ventana de acceso al sistema del administrador.	71
B.2.	Pantalla de inicio.	72
B.3.	Ventana donde se registran nuevos comercios.	72
B.4.	Ventana donde se registran nuevos usuarios.	73
B.5.	Informe de comercios.	74
B.6.	Ventana que permite modificar el comercio seleccionado.	74
B.7.	Informe de usuarios.	75
B.8.	Ventana que permite modificar el usuario seleccionado.	75
B.9.	Pantalla de <i>Login</i> .	76
B.10.	Pantalla de inicio.	77
B.11.	Formulario del perfil de comercio.	77
B.12.	Formulario de publicación de ofertas.	78
B.13.	Ventana de <i>Ofertas Activas</i> .	79
B.14.	Ventana de modificación de la oferta seleccionada.	79
B.15.	Ventana con las solicitudes de la oferta seleccionada.	80
B.16.	Historial de ofertas.	81
C.1.	Pantalla principal FI-Lab.	84
C.2.	Sección de <i>Floating IPs</i> .	85
C.3.	Reglas de acceso de ciertos servicios o puertos.	85
C.4.	Sección de <i>Keypairs</i> .	86

C.5.	Detalles técnicos de la instancia a lanzar.	87
C.6.	Asociación de <i>keypair</i> y <i>security groups</i> .	87
C.7.	Resumen de las características de la instancia.	88
D.1.	Código XML y JSON utilizado para crear una nueva entidad, con su posición.	90
D.2.	Respuesta de <i>Orion</i> al recibir una operación de tipo <i>updateContext</i> .	90
D.3.	Ejemplo de operación <i>queryContext</i> .	91
D.4.	Respuesta a la operación <i>queryContext</i> .	91
D.5.	Ejemplo de <i>queryContext</i> especial.	92
D.6.	Código XML y JSON de la suscripción de <i>Cygnus</i> en <i>Orion</i> .	93
D.7.	Ejemplo notificación guardada en <i>Cosmos</i> .	94
D.8.	Ejemplo de datos de la notificación guardados en <i>Cosmos</i> .	94
G.1.	Editor XML del entorno de programación utilizado para el diseño de la interfaz de usuario de <i>MyCity4Me</i> .	100
I.1.	Ejemplos de tipos de <i>layout</i> en Android.	103
I.2.	Sistema de comunicación entre Android y la base de datos.	106
I.3.	Métodos de interacción más comunes de Android. (a) Pulsación simple (b) Desplazamiento horizontal (c) Desplazamiento vertical (d) Zoom.	108
J.1.	Editor gráfico de DreamWeaver.	112
J.2.	Ejemplo de caja de texto y botón.	113

